

N73- 26884



LMSC-A991396

30 JUNE 1973

FINAL REPORT

SHUTTLE CRYOGENICS

SUPPLY SYSTEM

OPTIMIZATION STUDY

VOLUME V B-2

APPENDIX TO PROGRAMMERS MANUAL

FOR MATH MODEL

PART 1

CONTRACT NAS9-11330

CASE FILE
COPY

Prepared for Manned Spacecraft Center
by
Manned Space Programs, Space Systems Division

LOCKHEED MISSILES & SPACE COMPANY, INC.
A SUBSIDIARY OF LOCKHEED AIRCRAFT CORPORATION

FINAL REPORT
SHUTTLE CRYOGENIC SUPPLY SYSTEM
OPTIMIZATION STUDY

VOLUME VB-2
APPENDIX TO PROGRAMMERS MANUAL
FOR MATH MODEL

PART 1

Contract NAS 9-11330

Prepared for Manned Spacecraft Center By
Manned Space Programs, Space Systems Division

FOREWORD

This Final Report provides the results obtained in the Shuttle Cryogenics Supply System Optimization Study, NAS 9-11330, performed by Lockheed Missiles & Space Company (LMSC) under contract to the National Aeronautics and Space Administration, Manned Spacecraft Center, Houston, Texas. The study was under the technical direction of Mr. T. L. Davies, Cryogenics Section of the Power Generation Branch, Propulsion and Power Division. Technical effort producing these results was performed in the period from October 1970 to June 1973.

The Final Report is published in eleven volumes*:

Volume I	Executive Summary
Volumes II, III, and IV	Technical Report
Volumes VA-1 and VA-2	Math Model - Users Manual
Volumes VB-1, VB-2, VB-3, and VB-4	Math Model - Programmers Manual
Volume VI	Appendixes

The LMSC Staff participants are as follows:

Study Manager	L. L. Morgan
Subsystem Evaluations	C. J. Rudey D. P. Burkholder C. F. Merlet W. H. Brewington
Integrated Systems	H. L. Jensen
Component Analyses	B. R. Bullard F. L. Bishop
Thermodynamics	G. E. Heuer R. M. Vernon J. Gries D. R. Elgin

*The Table of Contents for all volumes appears in Volume I only. Section 12 in Volume III contains the List of References for Volumes I through IV.

Thermal Protection	G. E. Heuer R. Cima
Fluid Dynamics	D. P. Burkholder R. Cima
Propellant Acquisition	M. P. Hollister R. K. Grove
Design	R. A. Michael
Structural Analysis	M. L. Vaughn C. C. Richie
Instrumentation	R. R. Gaura
Reusability/Reliability	R. F. Hausman
Failure Modes and Effect Analyses	D. C. Saunders
Requirements and Criteria	C. F. Merlet
Safety and Mission Completion	C. F. Merlet
Math Model	R. F. Hausman J. McKay
<u>Cryogenic Cooling Subtask</u>	
Subsystem Evaluation	H. L. Jensen
Component Analyses	G. Heuer AiResearch
Thermodynamics	R. Cima
Thermal Protection	G. E. Heuer

CONTENTS




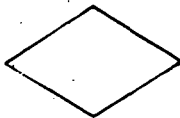




Appendix		Page
	FOREWORD	iii
A	Flow Chart Symbols	A-1
B	Program Listings	B-1

THE CRYOGENIC INTEGRATED
MATH MODEL PROGRAM
(TCIMM)

APPENDIX - A

FLOW CHART SYMBOLS

APPENDIX A
FLOW CHART SYMBOLS

<u>SYMBOL</u>	<u>DEFINITION</u>
	SUBPROGRAM REFERENCE
	PROCESSING FUNCTION
	PREPARATION FUNCTION
	DECISION FUNCTION
	TAPE OR FILE INPUT OR OUTPUT
	CARD INPUT
	FLOW DIRECTOR
	OFF-PAGE STEP CONNECTOR

Appendix B
THE CRYOGENIC INTEGRATED MATH MODEL
(TCIMM)

PART I - PROGRAM LISTING

The program listing presented in the following pages was produced using the EXEC-8 LISTALL processor which lists a file in alphabetical order. Since the processor does not differentiate between subroutines, functions and Procedure Definition Processors (PDPs), each subprogram has been relabeled to clearly identify the type of symbolic listing presented.

The alphabetical listing permits rapid list scanning when searching for a particular subprogram.

While symbolic listings are quite useful in understanding the coding of a particular subprogram, it is recommended that the program user create and maintain a standard compiler listing output file, since the additional information provided by the compiler is very useful in troubleshooting and debugging changes to the basic coding.

The program list file follows:

***** SUBROUTINE ACCRES

```

1  C      *****
2  C      * ROUTINE NAME - ACCUMULATOR RESIDUALS *
3  C      * DETERMINATION ROUTINE *
4  C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
5  C      * PROGRAMMER - R. BOLLINGER 1949 102 26933 *
6  C      * DATE CODED - 5/15/70 *
7  C      * REVISED - JULY 1972 *
8  C      * PROGRAMMER - J. MCKAY D1949 201 45178 *
9  C      *****
10 C
11 C      SUBROUTINE ACCRES
12 C
13 C      INCLUDE CACCUM
14 C
15 C      DO 100 IGS=1,2
16 C          FIND DENSITY OF GAS (RHOG)
17 C      CALL GSDNST (IGS,ATEMP(IGS),APRES(IGS),RHOG)
18 C          CALCULATE WEIGHT OF GAS RESIDUALS
19 C      WGRACC(IGS) = RHOG*AVOL(IGS)*NAOP(IGS)
20 C      100 CONTINUE
21 C
22 C      RETURN
23 C      END

```

SUBROUTINE ACQWT

```

1      C      * * * * *
2      C      * ROUTINE NAME - ACQUISITION SYSTEM WEIGHTS *
3      C      *      DETERMINATION ROUTINE      *
4      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
5      C      * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
6      C      * DATE CODED   - 5/20/70      *
7      C      * REVISED     - JULY 1972     *
8      C      * PROGRAMMER   - J. MCKAY D1943 201 45178 *
9      C      * * * * *
10     C
11     C      SUBROUTINE ACQWT
12     C
13     C      INCLUDE CCONFIG
14     C      INCLUDE CTANK
15     C
16     C      DIMENSION CST(2,3)
17     C
18     C      DATA (CST(1,1),I=1,2) / ,0035 ,.0333 /
19     C      DATA (CST(1,2),I=1,2) / ,00235,.01 /
20     C      DATA (CST(1,3),I=1,2) / ,0025 ,.007429 /
21     C
22     C      ***** THERE ARE 3 TYPES OF ACQUISITION DEVICES.
23     C      IDX = 1      SURFACE TENSION DEVICE
24     C      IDX = 2      POSITIVE DISPLACEMENT DEVICE
25     C      IDX = 3      DIELECTROPHORETIC DEVICE
26     C
27     C      DO 1000 I1 = 1,2
28     C      IDX = SATYPE(I1)
29     C
30     C      WTACQ(I1,1) = CST(1,IDX) * WPTOT(I1)
31     C      IDX = INDXTK(I1)
32     C      HEIGHT(IDX) = HEIGHT(IDX) + WTACQ(I1,1)
33     C      1000 CONTINUE
34     C      PRINT ACQUISITION SYSTEM WEIGHTS
35     C      CALL OTPACQ
36     C      RETURN
37     C      END

```

SUBROUTINE ALPHAB

```
1      SUBROUTINE ALPHAB(DB,DIB,TMODB)
2
3      C
4      C   CALCULATES ISOTHERMAL BULK MODULUS FROM THE EQUATION OF STATE
5      C   IN BRITISH UNITS
6      C   MUST FOLLOW CALLS OF PROPB AND DPDDB TO DEFINE D AND DI
7      C   INPUT AND OUTPUT PARAMETERS ARE IN BRITISH UNITS
8      C
9      TMODB=-DB*DIB
10     RETURN
11     END
```

SUBROUTINE APUFLO

```

1  SUBROUTINE APUFLO
2  E
3  INCLUDE CAPI
4  INCLUDE CCNTRL
5  INCLUDE CDCYCL
6  INCLUDE CENG
7  INCLUDE CIOUNT
8  INCLUDE CMATRL
9  INCLUDE TABLOK
10 C
11 C *****
12 C
13 C FIND PERCENT POWER FOR EACH DUTY CYCLE POINT
14 C
15 C   TOTHPR = HPR*NAFU
16 C   DO 200 IX = 1,KCYCLE
17 C     PCTHP(IX) = (100.0 * NEOP(IX) * HP(IX))/TOTHPR
18 C 200 CONTINUE
19 C
20 C *****
21 C
22 C CALCULATE PROPELLANT TEMPERATURE AT APU GAS GENERATOR INLET
23 C
24 C   IF(TIT - 2060.0) 205,205,210
25 C 205 TPF = (1.17 - FMR)/.000563
26 C   GO TO 215
27 C 210 TPF = (1.27 - FMR)/.000556
28 C
29 C *****
30 C
31 C SELECT THE PROPER COEFFICIENTS FOR COMPUTING THE REFERENCE PROPELLANT
32 C FLOW RATE.
33 C
34 C 215 IF(PGG - 900.0) 216,220,216
35 C 216 IF(PGG - 600.0) 217,225,217
36 C 217 IF(PGG - 300.0) 218,230,218
37 C
38 C ERROR IN THE SPECIFICATION OF PGG.
39 C
40 C 218 WRITE(6,25) PGG
41 C 25 FORMAT(10,'PGG IS SPECIFIED INCORRECTLY. PGG = ',F8.2, '//)
42 C   CALL EXIT
43 C 220 M = 0
44 C   RRFP = 8.70
45 C   GO TO 240
46 C 225 M = 1
47 C   RRFP = 9.30
48 C   GO TO 240
49 C 230 M = 2
50 C   RRFP = 10.57
51 C 240 CONTINUE
52 C
53 C *****
54 C
55 C COMPUTE THE REFERENCE PROPELLANT FLOW RATE IN POUNDS PER MINUTE AND
56 C THE CORRECTION FACTOR FOR THE REFERENCE PROPELLANT FLOW RATE WHEN
57 C TIT = 2060 DEGREES R.

```



```

*****
APUFLO *****
58 C
59 C1 = 1.0 + FMR
60 C2 = 1.0 + (1.0/FMR)
61 WOT = 0.0
62 WTBURN = 0.0
63 TIPWT = 0.0
64 WDOT = 0.0
65 I = 0
66 DO 260 J1 = 1,DCYCL,2
67 I = I + 1
68 IF(DCYCLE(J1)) ,260,
69 C
70 C LOOK UP RR VS PCTHP
71 C
72 CALL FINTAB (NTBID(3))
73 XTAB(1) = PGG
74 XTAB(2) = FMR
75 XTAB(3) = PAMB(I)
76 XTAB(4) = PCTHP(I)
77 RR(I) = MIPE(4,XTAB)
78 C
79 C CHECK TO SEE IF CORRECTION FACTOR IS NECESSARY
80 C
81 IF(TIT - 2060.0) 250,245,250
82 C
83 C LOOK UP CORRECTION FACTOR KK VS PCTHP
84 C
85 245 CALL FINTAB (NTBID(4))
86 KK(I) = MIPE(4,XTAB)
87 RR(I) = RR(I) * KK(I)
88 250 CONTINUE
89 C
90 C *****
91 C
92 C COMPUTE THE TOTAL PROPELLANT FLOW RATE OVER EACH CONSTANT POWER TIME
93 C INTERVAL IN LBS/MIN, AND COMPUTE THE TOTAL PROPELLANT USED OVER THE
94 C ENTIRE DUTY CYCLE.
95 C
96 WD(I) = (TOTHPR * RR(I))/300.0
97 WDOT1 = WD(I)
98 TIPWT = TIPWT + WDOT1 * DCYCLE(J1)
99 WDOTJ(1,2) = (WDOT1/C1)/60.0
100 WDOTJ(1,1) = WDOT1/60.0 - WDOTJ(1,2)
101 IF(NEOP(I).LT.0) GO TO 260
102 WDOT = AMAX1(WDOT,WDOT1)
103 WTBURN = WTBURN + WD(I) * DCYCLE(J1)
104 C
105 C SET CONTINGENCY QUANTITY OF TOTAL PROPELLANT FOR RESERVE
106 C
107 WDT = 1.3 * WTBURN
108 C
109 C CALCULATE THE WEIGHT RATE OF H2 AND O2 FLOWING DURING THE TIME PERIOD
110 C THETA(I) IN LBS/MIN.
111 C
112 WDRH(I) = WD(I)/C1
113 WDRO(I) = WD(I)/C2
114 260 CONTINUE
115 C

```

```

*****  APUFLO  *****
116  C  CALCULATE THE TOTAL H2 AND O2 DELIVERED TO APU GAS GENERATOR IN LBS.
117  C
118      WDH = WDT/C1
119      WDOTI(2) := (WDOT/C1)/60.0
120  C
121      WDO = WDT/C2
122      WDOTI(1) := WDOT/60.0 - WDOTI(2)
123  C
124  C  *****
125  C
126  C  DETERMINE APU EXHAUST GAS TEMPERATURE DURING POWER TIME INTERVAL
127  C  THETA(I). FIRST COMPUTE THE SPECIFIC HEAT CAPACITY OF THE COMBUSTION
128  C  PRODUCTS AS A FUNCTION OF MIXTURE RATIO AND TURBINE INLET TEMPERATURE
129  C
130      CALL CSUBPI(TIT,FMR,CP)
131      DO 405 I = 1,KCYCLE
132          TE(I) = TIT - ((42.42 * NEOP(I) * HP(I)) / (WD(I) * CP))
133          TME = (TE(I) + TD)/2.0
134  C
135  C  COMPUTE THE SPECIFIC HEAT OF COMBUSTION PRODUCTS AS A FUNCTION OF
136  C  MIXTURE RATIO AND TEMPERATURE OF EXHAUST GASES.
137  C
138      CALL CSUBPI(TME,FMR,CPE(I))
139      D(I) = CPE(I)*(TE(I) - TD)
140 405 CONTINUE
141  C
142  C  *****
143  C
144  C  ***** OUTPUT THE DATA *****
145  C
146      CALL OPAPUF (KCYLE)
147  C
148      RETURN
149  C
150  END

```

SUBROUTINE APUSUB

```

1      SUBROUTINE APUSUB
2      C
3      C
4      INCLUDE CACCUH
5      INCLUDE CAPU
6      INCLUDE CDCYCL
7      INCLUDE CENG
8      INCLUDE CLOUNT
9      INCLUDE CPUMP
10     INCLUDE CTANK
11     C
12     C *****
13     C
14     C BEGIN COMPUTATIONS FOR SUBCRITICAL STORAGE
15     C
16     C *****
17     C
18     C SET SELECTED VARIABLES FROM INPUT DATA
19     C
20     TAH = ATEMP(2)
21     TAO = ATEMP(1)
22     TSTH = SITEHP(2,1)
23     TSTO = SITEHP(1,1)
24     PSTH = SOPRES(2,1)
25     PSTO = SOPRES(1,1)
26     C
27     C *****
28     C
29     C SIZE HEAT EXCHANGER BETWEEN H2 ACCUMULATOR AND APU GAS GENERATOR
30     C
31     CALL CSUBP(TPF,PGG,2,CPSAH)
32     CALL CSUBP(TPF,PGG,1,CPSAO)
33     C1 = CPSAH*(TPF - TAH)
34     C2 = CPSAO*(TPF - TAO)
35     DO 610 I = 1,KCYCLE
36     Q4HDOT(I) = WDRH(I)*C1
37     Q6ODOT(I) = WDRO(I)*C2
38     HDG(I) = Q4HDOT(I)/D(I)
39     610 WDJ(I) = Q6ODOT(I)/D(I)
40     C
41     C *****
42     C
43     C
44     C COMPUTE THE TEMPERATURES OF THE EXHAUST GASES FROM THE H2 AND O2
45     C CONDITIONING GAS GENERATORS.
46     C
47     TGGCH = (MRGGCH + .00056*TAH + .056)*1000.0/.591
48     TGGCO = (MRGGCO + .00056*TAH + .056)*1000.0/.591
49     C
50     C *****
51     C
52     C SET THE TEMPERATURE OF THE GASES AT DISCHARGE FROM THE PUMP EQUAL TO
53     C THE TEMPERATURE OF THE GAS IN THE STORAGE TANK.
54     C
55     TPDH = TSTH
56     TPDO = TSTO
57     C

```

```

*****
APUSUB *****
58 C *****
59 C
60 C SIZE H2 AND O2 HEAT EXCHANGER BETWEEN THE PUMPS AND ACCUMULATORS
61 C
62     HAH = HYENTH(PPDCH(2),TAH)
63     HPH = HYENTH(PPDCH(2),TPDH)
64     HAO = OXENTH(PPDCH(1),TAO)
65     HPO = OXENTH(PPDCH(1),TPDO)
66     TME = (TGGCO + TDGGO)/2.0
67     CALL CSUBP1(TME,MRGGCO,CPGGO)
68     C1 = (1.0 + MRGGCO)*CPGGO*(TGGCO - TDGGO)
69     A1 = (HAO - HPO)*MRGGCO/C1
70     A2 = (HAO - HPO)/C1
71     TME = (TGGCH + TDGGH)/2.0
72     CALL CSUBP1(TME,MRGGCH,CPGGH)
73     C2 = (1.0 + MRGGCH)*CPGGH*(TGGCH - TDGGH)
74     A4 = (HAH - HPH)/C2
75     A3 = A4*MRGGCH
76     C1 = (A1 + A4) + (A2*A3) - (A1*A4) - 1.0
77     C2 = (A1*A4) - (A2*A3) - A4
78     C3 = (A1*A4) - (A2*A3) - A1
79     DO 650 I = 1,KCYCLE
80     WGGH(I) = (WDRH(I)*C2 - WDRO(I)*A2)/C1
81     WGGO(I) = (WDRO(I)*C3 - WDRH(I)*A3)/C1
82     QSHDOT(I) = (WDRH(I) + WGGH(I))*(HAH - HPH)
83     QTODOT(I) = (WDRO(I) + WGGO(I))*(HAO - HPO)
84     650 CONTINUE
85 C
86 C *****
87 C
88 C COMPUTE THE TOTAL WEIGHT OF H2 AND O2 NEEDED FOR THE H2 AND O2
89 C CONDITIONING GAS GENERATORS.
90 C
91     WTGGH = 0.0
92     WTGGO = 0.0
93     I = 0
94     DO 660 II = 1,NDCYCL*2
95     I = I + 1
96     WTGGH = WTGGH + WGGH(I) * DCYCLE(II)
97     WTGGO = WTGGO + WGGO(I) * DCYCLE(II)
98     660 CONTINUE
99 C
100 C *****
101 C
102 C CALCULATE THE VOLUMES OF THE H2 AND O2 STORAGE TANKS.
103 C
104     CALL RHOLIQ(TSTH,2,RHOLH)
105     CALL RHOLIQ(TSTO,1,RHOLo)
106 C
107 C
108     CALL GSDNST (2,TSTH,PSTH,RHOGH)
109     CALL GSDNST (1,TSTO,PSTO,RHOGO)
110 C
111 C ***
112 C WEIGHT OF H2 REQD. TO ABSORB H2 TANK HEATLEAK
113 C ***
114 C
115     TAUsum = 0.0

```

```

***** APUSUB *****
116 DO 661 I1 = 2, NDCYCL, 2
117 IF (DCYCLE(I1)) 661,
118 TAUSUM = TAUSUM + DCYCLE(I1+1)
119 661 CONTINUE
120 TAUSUM = TAUSUM/60.
121 VESTH = ((WDH + WTGGH)/(RHOLH-RHOGH))*1.03
122 AREATH = 4.84 * (VESTH**0.667)
123 TSATLV = PSATH(PSTH, HG, HL)
124 DELHV = (HG - HL)
125 CALL TCOND(TENV, TSTH, SNBAR(2), SITHIK(2,1), SITYPE(2,1), QTHDOT)
126 WSVHH = (1.0/DELHV)*(QTHDOT*AREATH*TAUSUM)
127 C ***
128 C WEIGHT OF H2 REQD. TO ABSORB O2 TANK HEATLEAK
129 C ***
130 DELTH2 = (TSTO - TSTH)
131 CALL CSUBP(TSTO, PSTO, 2, CPH1)
132 CALL CSUBP(TSTH, PSTH, 2, CPH2)
133 AVGCPH = (CPH1 + CPH2)/2.0
134 DELHDT = DELTH2 * AVGCPH
135 VESTO = ((WDO + WTGGO)/(RHOLO-RHOGO))*1.03
136 AREATO = 4.84 * (VESTO**0.667)
137 CALL TCOND(TENV, TSTO, SNBAR(1), SITHIK(1,1), SITYPE(1,1), QTODOT)
138 WSVHO = (1.0/DELHDT)*(QTODOT*AREATO*TAUSUM)
139 IF (WSVHH.GE.WSVHO) WSVH = WSVHH
140 IF (WSVHH.LT.WSVHO) WSVH = WSVHO
141 C
142 VSTH = ((WDH+WTGGH+WSVH)/(RHOLH-RHOGH))*1.03
143 AREATH = 4.84 * (VSTH**0.667)
144 VSTO = ((WDO+WTGGO)/(RHOLO-RHOGO))*1.03
145 C
146 WPTOT(2) = WDH + WTGGH + WSVH
147 WPTTH = WPTOT(2)
148 WPTOT(1) = WDO + WTGGO
149 WPTTO = WPTOT(1)
150 C
151 C *****
152 C
153 C CALCULATE THE WIEGHT OF THE H2 AND O2 STORAGE TANK RESIDUAL
154 C PROPELLANTS.
155 C
156 C
157 WSRH = RHOGH * VSTH
158 WSR0 = RHOGO * VSTO
159 C
160 C *****
161 C
162 C CALCULATE THE WEIGHT OF H2 AND O2 ACCUMULATOR RESIDUAL PROPELLANTS.
163 C
164 CALL ZFIND(TPF, PPDCH(2), 2, ZSAH)
165 CALL ZFIND(TAH, PGG, 2, ZSAHE)
166 ZSAO = ZGET(TPF, PPDCH(1), 1)
167 ZSAOE = ZGET(TAO, PGG, 1)
168 C1 = (PPDCH(2)/ZSAH) - (PGG/ZSAHE)
169 C2 = (PPDCH(1)/ZSAO) - (PGG/ZSAOE)
170 C CALC. CAPICITY OF ACCUM.
171 WSHB = (HPR*RRFP)/(18000.0*(1.0+FMR))
172 WSOB = (HPR*RRFP)/(18000.0*(1.0+(1.0/FMR)))
173 C

```

***** APUSUB *****

```

174      WRSAB = (WSHB*PGG)/(C1*ZSAHE)
175      WRSAB = (WSOB*PGG)/(C2*ZSAOE)
176      C
177      C *****
178      C
179      C      ***** ACCUMULATOR VOLUME IS INPUT *****
180      C
181      C *****
182      C
183      C *****
184      C
185      C      OUTPUT THE DATA
186      C
187      C *****
188      C
189      C      CALL OAPUSB (KCYCLE)
190      C
191      C      WRITE (IOT,999)
192      C      999 FORMAT(////T30,'***** THE APU SUBCRITICAL CALCULATIONS HAVE BEEN
193      C      1 COMPLETED *****')
194      C
195      C
196      C      667 CONTINUE
197      C
198      C      RETURN
199      C
200      C      END

```

***** SUBROUTINE APUSUP

```

1      SUBROUTINE APUSUP
2      C
3      C
4      REAL K7, K8, K11
5      C
6      INCLUDE CACCUM
7      INCLUDE CAPU
8      INCLUDE CCTRL
9      INCLUDE CDCYCL
10     INCLUDE CENG
11     INCLUDE CFUEL
12     INCLUDE CHEX
13     INCLUDE CIOUNT
14     INCLUDE CMATRL
15     INCLUDE CONST
16     INCLUDE CPIMP
17     INCLUDE CTANK
18     INCLUDE TABLOK
19     C
20     DIMENSION WDOTX(MHX,2)
21     C
22     EQUIVALENCE (WVHO2,WVIHO), (WVHH2,WVIHH)
23     EQUIVALENCE (WDOTCF,WDOTX)
24     C
25     C *****
26     C
27     BEGIN COMPUTATIONS FOR SUPERCRITICAL STORAGE
28     C
29     C *****
30     C
31     SET SELECTED VARIABLES FROM INPUT DATA
32     C
33     TAH = ATEMP(2)
34     TAO = ATEMP(1)
35     PCH = SOPRES(2,1)
36     PCO = SOPRES(1,1)
37     C
38     C *****
39     C
40     SIZE H2 HEAT EXCHANGER BETWEEN H2 ACCUMULATOR AND APU GAS GENERATOR.
41     C
42     406 HPH = HYENTH(PCH,TPF)
43     HAH = HYENTH(PCH,TAH)
44     HPO = OXENTH(PCO,TPF)
45     HAO = OXENTH(PCO,TAO)
46     DO 410 I = 1,KCYCLE
47     QIHDOT(I) = WDRH(I)*(HPH - HAH)
48     WDA(I) = QIHDOT(I)/D(I)
49     C
50     C *****
51     C
52     C
53     SIZE O2 HEAT EXCHANGER BETWEEN O2 ACCUMULATOR AND APU GAS GENERATOR.
54     C
55     QIODOT(I) = WDRO(I)*(HPO - HAO)
56     WDO(I) = QIODOT(I)/D(I)
57     410 CONTINUE

```

```

***** APUSUP *****
58 C
59 C
60 C *****
61 C
62 C
63 C DETERMINE INITIAL TANK TEMPERATURES
64 C
65 CALL FINTAB(NTBID(8))
66 XTAB(1) = PCO
67 XTAB(2) = 70.126
68 TEMPO2 = MIPE(2,XTAB)
69 C
70 CALL FINTAB(NTBID(7))
71 XTAB(1) = PCH
72 XTAB(2) = 4.365
73 TEMPH2 = MIPE(2,XTAB)
74 C
75 C DETERMINE INITIAL CSUBV VALUES FOR TANK T AND P CONDITIONS
76 C
77 CISBVO = CSUBV(TEMPO2,PCO,1)
78 CISBVH = CSUBV(TEMPH2,PCH,2)
79 C
80 C COMPUTE THE COMPRESSIBILITY OF H2 AT TEMPERATURE TFH AND PRESSURE PFH
81 C
82 C COMPUTE THE COMPRESSIBILITY OF O2 AT TEMPERATURE TFO AND PRESSURE PFO
83 C
84 TTTH = 0.0
85 TTTO = 0.0
86 TTH2WD(1) = 0.0
87 TTO2WD(1) = 0.0
88 I = 0
89 DO 450 I = 1,NDCYCL,2
90 I = I + 1
91 C
92 CALL ZFIND(TFH,PFH,2,ZFH)
93 C
94 ZFO = ZGET(TFO,PFO,1)
95 C
96 C COMPUTE THE PERCENT OF USABLE H2 AND O2 WITHDRAWN UP TO THIS POINT
97 C IN THE MISSION
98 C
99 TTTH = TTTH + DCYCLE(I) * WDRH(I)
100 TTH2WD(I) = TTTH
101 TTTO = TTTO + DCYCLE(I) * WDRO(I)
102 TTO2WD(I) = TTTO
103 PCH2WD(I) = TTH2WD(I)/WDH
104 PCO2WD(I) = TTO2WD(I)/WDO
105 C
106 C COMPUTE THE DENSITY OF H2 AND O2 AS A FUNCTION OF PERCENT WITHDRAWN.
107 C
108 C1 = 1.0 - ((0.0427*PFH)/(ZFH*TFH))
109 C2 = 1.0 - ((0.04253*PFO)/(ZFO*TFO))
110 RHOCH2(I) = 4.365 * (1.0 - (PCH2WD(I) * C1))
111 RHOCH2(I) = 70.126 * (1.0 - (PCO2WD(I) * C2))
112 C
113 C *****
114 C
115 C COMPUTE THE TEMPERATURE OF H2 IN STORAGE TANK DURING TIME INTERVAL

```



```

***** APUSUP *****
16 C THETA(I) AS A FUNCTION OF DENSITY AND CONDITIONED PRESSURE.
17 C
18 C CALL FINTAB (NTBID(7))
19 C XTAB(1) = PCH
20 C XTAB(2) = RHOC2(I)
21 C TTH(I) = MIPE(2,XTAB)
22 C
23 C COMPUTE THE TEMPERATURE OF O2 IN STORAGE TANK AS A FUNCTION OF
24 C DENSITY AND PRESSURE.
25 C
26 C CALL FINTAB (NTBID(8))
27 C XTAB(1) = PCO
28 C XTAB(2) = RHOC2(I)
29 C TTO(I) = MIPE(2,XTAB)
30 C
31 C *****
32 C
33 C COMPUTE SPECIFIC HEAT OF H2 AS A FUNCTION OF DENSITY AND STORED
34 C PRESSURE.
35 C
36 C CALL FINTAB (NTBID(5))
37 C XTAB(1) = PCH
38 C XTAB(2) = RHOC2(I)
39 C DQDWH(I) = MIPE(2,XTAB)
40 C
41 C COMPUTE THE SPECIFIC HEAT OF O2 AS A FUNCTION OF DENSITY AND PRESSURE
42 C
43 C CALL PHTON(TTO(I),RHOC2(I),1,PHI,THETA)
44 C DQDWO(I) = THETA
45 C
46 C *****
47 C
48 C
49 C COMPUTE THE REQUIRED FLOW RATE OF EXHAUST GASES THROUGH HEAT EXCHANGER
50 C BETWEEN H2 TANK AND H2 ACCUMULATOR.
51 C
52 C HTH(I) = HYENTH(PCH,TTH(I))
53 C Q2HDOT(I) = WDRH(I) * ( HAH - HTH(I))
54 C WDB(I) = Q2HDOT(I)/D(I)
55 C HTO(I) = OXENTH(PCO,TTO(I))
56 C
57 C DO THE SAME FOR THE HEAT EXCHANGER BETWEEN THE O2 TANK AND O2 ACCUMU-
58 C LATOR.
59 C
60 C Q2ODOT(I) = WORO(I) * ( HAO - HTO(I))
61 C WDE(I) = Q2ODOT(I)/D(I)
62 C
63 C SIZE H2 TANK HEAT EXCHANGER AND O2 TANK HEAT EXCHANGER.
64 C
65 C Q3HDOT(I) = WDRH(I)*DQDWH(I)
66 C WDC(I) = Q3HDOT(I)/D(I)
67 C Q3ODOT(I) = WORO(I)*DQDWO(I)
68 C WDF(I) = Q3ODOT(I)/D(I)
69 C
70 C CSUBVO(I) = CSUBV(TTO(I),PCO,1)
71 C CSUBVH(I) = CSUBV(TTH(I),PCH,2)
72 C
73 C CALL PHTON(TTO(I),RHOC2(I),1,PHI,THETA)

```

```

***** APUSUP *****
174      PHI2(I) = PHI
175      C
176      CALL FINTAB (NTBID(40))
177      XTAB(1) = PCH
178      XTAB(2) = RHOC2(I)
179      PHI2(I) = HIPE(2,XTAB)
180      C
181      450 CONTINUE
182      C
183      C *****
184      C
185      C
186      C CHECK ADEQUACY OF EXHAUST PRODUCTS FOR CONDITIONING THE GASES
187      C
188      DO 500 I = 1,KCYCLE
189      WSUM(I) = WDA(I) + WDB(I) + WDC(I) + WDD(I) + WDE(I) + WDF(I)
190      DWDB(I) = 0.0
191      WGHC(I) = 0.0
192      WGOI(I) = 0.0
193      DWDB(I) = WSUM(I) - WD(I)
194      500 CONTINUE
195      C
196      DO 501 I = 1,KCYCLE
197      IF (WSUM(I) .LE. WD(I)) GO TO 501
198      GO TO 460
199      501 CONTINUE
200      GO TO 470
201      C
202      C
203      C *****
204      C
205      C
206      C CORRECT FOR THE EFFECTS OF USE OF THE SUPPLEMENTARY CONDITIONING GAS
207      C GENERATOR AND HEAT EXCHANGER.
208      C
209      460 DO 502 I = 1,KCYCLE
210      TME = (TG + TD)/2.0
211      CALL CSUBPI(TME,FMRG,CPG)
212      K7 = CPG*(TG - TD)
213      KB = 1.0 + FMRG
214      K11 = (WDC(I)/WDRH(I)) + ((WDE(I)*FMRG)/WDRO(I)) + ((WDF(I)*FMRG)/
215      I /WDRO(I))
216      C
217      C COMPUTE REFERENCE H2 AND O2 FLOWRATES TO SUPPLEMENTAL GAS GENERATOR.
218      C
219      C WGH(I) = (WD(I) - WSUM(I))/(K11 - ((KB*K7)/D(I)))
220      C
221      C
222      C WGH(I) = WGH(I)
223      C
224      C WGOI(I) = FMRG*WGH(I)
225      C
226      C COMPUTE THE CORRECTED VALUES OF THE HEAT EXCHANGER EXHAUST REQUIREMEN
227      C TS.
228      C
229      WDBC(I) = WDB(I) - (WGH(I)*((KB*K7)/D(I)))
230      WDCC(I) = WDC(I) * (1.0 + (WGH(I)/WDRH(I)))
231      WDEC(I) = WDE(I) * (1.0 + (WGOI(I)/WDRO(I)))

```

```

***** APUSUP *****
232      WDFC(I) = WDF(I) * (1.0 + (WGOC(I)/WDRG(I)))
233      502 CONTINUE
234      C
235      GO TO 471
236      ***
237      C IF NO SUPPLEMENTARY CONDITIONING GAS GENERATOR AND HEAT EXCHANGER
238      C ARE REQUIRED.
239      C ***
240      470 DO 503 I = 1,KCYCLE
241          WDBC(I) = WDB(I)
242          WDC(I) = WDC(I)
243          WDEC(I) = WDE(I)
244          WDFC(I) = WDF(I)
245      503 CONTINUE
246      C
247      471 CONTINUE
248      C
249      C COMPUTE THE CORRECTED TOTAL H2 AND O2 FLOWS TO THE H2 AND O2 ACCUMULA
250      C TORS.
251      C
252      DO 504 I = 1,KCYCLE
253          WTH(I) = WDRH(I) + WGHG(I)
254          WTO(I) = WDRG(I) + WGOC(I)
255      C
256      C COMPUTE THE TOTAL ENTHALPY INCREMENT SUPPLIED BY SUPPLEMENTARY GAS.
257      C
258      DELH(I) = WGHG(I) * K8 * K7
259      C
260      C COMPUTE CORRECTED VALUES OF HEAT FLOW IN MAIN H2 AND O2 HEAT EXCHANG-
261      C ERS BETWEEN TANK AND ACCUMULATOR.
262      C
263      Q2HDTG(I) = D(I)*WDBC(I)
264      Q2ODTG(I) = WTO(I)*(H4O - HTO(I))
265      C
266      C COMPUTE THE CORRECTED VALUES OF THE HEAT FLOWS IN THE H2 AND O2 TANK
267      C HEAT EXCHANGERS.
268      C
269      Q3HDTG(I) = DQODWH(I)*WTH(I)
270      Q3ODTG(I) = DQODWO(I)*WTO(I)
271      C
272      C COMPUTE TEMPERATURE OF COLD FLUID ENTERING SUPPLEMENTARY GAS
273      C GENERATOR
274      C
275      CALL CSUBP(TAH,PCH,2,CPH(I))
276      C
277      CALL CSUBP(TTH(I),PCH,2,CPBH(I))
278      C
279      TSIN(I) = TAH - (((WGHG(I)+WGOC(I))*K7)/(WTH(I)*CPBH(I)))
280      C
281      504 CONTINUE
282      C
283      DO 505 I4 = 1,KCYCLE
284          IF(TSIN(I4) .LT. TTH(I4)) GO TO 506
285      505 CONTINUE
286      C
287      GO TO 507
288      C
289      506 CONTINUE

```

```

***** APUSUP *****
290 C
291 C *** RESET VALUE OF FMR AND RECYCLE PROGRAM.
292 C
293 FMR = FMR + 0.1
294 LREPT = 1
295 C
296 WRITE (107,7011) I4, TSIN(I4), FMR
297 7011 FORMAT(///T20,100(' '),T20,100(' '),T20,100(' '),T27,1AN UNACCEPT
298 IABLE VALUE FOR TSIN HAS BEEN ENCOUNTERED - - - TSIN FOR CYCLE =',
299 213,2X, ' ',F8.2//T40,1 THE APU FUEL MIXTURE RATIO HAS BEEN RESET T
300 30 - - - FMR =',F5.2//T20,100(' '),T20,100(' '),T20,100(' '))
301 C
302 RETURN
303 C
304 507 CONTINUE
305 C
306 C *****
307 C
308 C
309 C CALCULATE THE WEIGHT OF PROPELLANT TANK HEATER CIRCULATING COMPRESSOR
310 C FIRST COMPUTE THE MAXIMUM FLOW RATE FOR HYDROGEN AND OXYGEN.
311 C
312 DGWMXO = 0.0
313 DOWMXH = 0.0
314 DO 509 I = 1,KCYCLE
315 DOWMXO = AMAXI(DGWMXO,DGODWO(I))
316 DOWMXH = AMAXI(DOWMXH,DGODWH(I))
317 509 CONTINUE
318 C
319 WOMAX = 0.0
320 WHMAX = 0.0
321 DO 510 I = 1,KCYCLE
322 WOMAX = AMAXI(WOMAX,WTO(I))
323 WHMAX = AMAXI(WHMAX,WTH(I))
324 510 CONTINUE
325 C
326 DO 511 I = 1,KCYCLE
327 QINTKO(I) = DGODWO(I) * WTO(I)
328 QINTKH(I) = DGODWH(I) * WTH(I)
329 511 CONTINUE
330 C
331 QMXTKO = 0.0
332 QMXTKH = 0.0
333 DO 512 I = 1,KCYCLE
334 QMXTKO = AMAXI(QMXTKO,QINTKO(I))
335 QMXTKH = AMAXI(QMXTKH,QINTKH(I))
336 512 CONTINUE
337 C
338 C
339 C COMPUTE THE SPECIFIC HEAT AND DENSITY OF H2 AND O2 AT THE FINAL
340 C TEMPERATURE AND PRESSURE
341 C
342 CALL CSUBP(TFO,PFO,1,CPFO)
343 CALL DENSON(TFO,PFO,1,RHOOF,ZEE)
344 C
345 IGAS = 2
346 CALL CSUBP(TFH,PFH,IGAS,CPFH)
347 RHOMF = (144.0*PFH)/(ZFH*TFH*FINDR(IGAS))

```

```

***** APUSUP *****
348 C
349 WDTCP0 = QMXTKO/(CPFO * (TD-100.0-TFO))
350 WDTCPH = QMXTKH/(CPFH * (TD-100.0-TFH))
351 C
352 WOCOMP = (0.01455*DELPCP*WDTCP0)/RHOOF
353 WCIRCP(1) = WOCOMP
354 WHCOMP = (0.01455*DELPCP*WDTCPH)/RHOHF
355 WCIRCP(2) = WHCOMP
356 C
357 C CALCULATE THE WEIGHT OF THE H2 AND O2 STORAGE TANKS. FIRST COMPUTE
358 C THE TOTAL WEIGHT OF H2 AND O2 NEEDED FOR THE SUPPLEMENTAL GAS
359 C GENERATOR.
360 C
361 WTGH = 0.0
362 WTGO = 0.0
363 I = 0
364 DO 520 I1 = 1, NDCYCL*2
365 I = I + 1
366 WTGH = WTGH + WGHC(I) * DCYCLE(I1)
367 WTGO = WTGO + WGOC(I) * DCYCLE(I1)
368 520 CONTINUE
369 C
370 C *****
371 C
372 C
373 C BEGIN CALCULATING WEIGHT OF H2 VENTED DURING THE MISSION. FIRST
374 C COMPUTE THE APPROXIMATE VOLUME OF THE H2 AND O2 STORAGE TANKS.
375 C
376 VTH = (WDH + WTGH)/(4.35 * RHOHF)
377 VTO = (WDO + WTGO)/(70.0 * RHOOF)
378 RH = CBRT((3.0*VTH)/12.566)
379 RO = CBRT((3.0*VTO)/12.566)
380 C
381 C COMPUTE THE AREA OF THE H2 AND O2 TANKS ASSUMING THAT THEY HAVE
382 C SPHERICAL GEOMETRY.
383 C
384 ATH = 12.566*(RH**2)
385 ATO = 12.566*(RO**2)
386 C
387 C COMPUTE THE HEAT FLOW PER UNIT AREA PER UNIT TIME ACROSS THE SURFACE
388 C OF THE TANK MATERIALS.
389 C
390 WVIHH = 0.0
391 WVIHO = 0.0
392 I = 0
393 DO 550 I1 = 2, NDCYCL*2
394 I = I + 1
395 C
396 C CALL SUBROUTINE TCOND TO RETURN QDOTH.
397 C
398 CALL TCOND(TENV,TFH,SNBAR(2),SITHIK(2,1),SITYPE(2,1),QDOTH)
399 DHH = (QDOTH * ATH * DCYCLE(I1))/60.0
400 C
401 CALL TCOND(TENV,TFO,SNBAR(1),SITHIK(1,1),SITYPE(1,1),QDOTO)
402 DHO = (QDOTO * ATO * DCYCLE(I1))/60.0
403 C
404 C COMPUTE THE SPECIFIC HEAT OF H2 AT T= TTH(1) AND P= PCH.
405 C

```

```

***** APUSUP *****
406      CALL CSUBP(TTH(I),PCH,2,CPH)
407      WVIHO = WVIHO + (DHO/(CPH*(TFO-TTH(I))))
408      C
409      C COMPUTE THE WEIGHT OF THE H2 IN THE TANK AT THE START OF THE COAST
410      C PERIOD TAU(I).
411      C
412      WH = WDH + WTGH - TTH2WD(I)
413      C1 = (5.4*DHH)/(PCH*VTH)
414      C2 = SQRT(1.0 + C1**2)
415      WVIHH = WVIHO + (WH * (1.0+C1-C2))
416      550 CONTINUE
417      C
418      C COMPUTE RESULTANT TOTAL WEIGHT OF VENTED H2.
419      C
420      IF(WVIHH-WVIHO) ,555,555
421      WVH = WVIHO
422      IF(WVH.LT.0.0) WVH = 0.0
423      GO TO 560
424      555 WVH = WVIHH
425      560 CONTINUE
426      C
427      C *****
428      C
429      C
430      C CALCULATE WEIGHT OF H2 AND O2 STORAGE TANKS. FIRST COMPUTE THE
431      C CORRECTED VOLUME OF THE H2 STORAGE TANK
432      C
433      VTH = (WDH + WTGH + WVH)/(4.35 - RHOHF)
434      MTRL = SMTYPE(2,1)
435      CALL FINTAB (NTBID(9)+MTRL)
436      FTUX = MIPE (1,TFH)
437      ROFTUH = RHOL(MTRL)/1728./FTUX
438      MTRL = SMTYPE
439      CALL FINTAB (NTBID(9)+MTRL)
440      FTUX = MIPE (1,TFO)
441      ROFTUD = RHOL(MTRL)/1728./FTUX
442      WGTHT = 7000.0*VTH*PCH*ROFTUH
443      WGTOT = 7000.0*VTO*PCO*ROFTUD
444      C
445      SMDIAM(2,1) = ((6.0 * VTH)/PI)**0.333
446      C
447      SMDIAM(1,1) = ((6.0 * VTO)/PI)**0.333
448      C
449      C *****
450      C
451      C
452      C CALCULATE WEIGHT OF RESIDUAL PROPELLANTS IN THE H2 AND O2 TANKS
453      C
454      WRH = VTH*RHOHF
455      WRO = VTO*RHOOF
456      C
457      C *****
458      C
459      C CALCULATE WEIGHT OF H2 AND O2 ACCUMULATOR TANKS
460      C
461      MTRL = AMTYPE(2)
462      CALL FINTAB (NTBID(9)+MTRL)
463      FTUX = MIPE (1,TAH)

```

```

*****  APUSUP  *****
464      ROFTUH = RHOL(MTRL)/1728./FTUX
465      MTRL = AMTYPE
466      CALL FINTAB (NTBID(9)+MTRL)
467      FTUX = MIPE (1,TAO)
468      ROFTUO = RHOL(MTRL)/1728./FTUX
469      CALL ZFIND(TAH,PCH,2,ZAH)
470      CALL ZFIND(TAH,PFH,2,ZAHF)
471      ZAO = ZGET(TAO,PCO,1)
472      ZAOF = ZGET(TAO,PFO,1)
473      C1 = (PCH/ZAH) = (PFH/ZAHF)
474      C2 = (PCO/ZAO) = (PFO/ZAOF)
475      C
476      WAH = ((2.085/(1.0+FMR))*{PCH*ROFTUH}*{HPR*RRFP*TAH})/C1
477      C
478      WAO = ((0.1308/(1.0+(1.0/FMR))*{PCO*ROFTUO}*{HPR*RRFP*TAO})/C2
479      C
480      C
481      C CALCULATE THE WEIGHT OF THE H2 AND O2 ACCUMULATOR RESIDUAL PROPELLANT
482      C
483      WSH = (HPR*RRFP)/(18000.0*(1.0+FMR))
484      WSO = (HPR*RRFP)/(18000.0*(1.0+(1.0/FMR)))
485      WRAH = (WSH*PFH)/(C1*ZAHF)
486      WRAO = (WSO*PFO)/(C2*ZAOF)
487      C
488      C
489      C *****
490      C
491      C COMPUTE THE TOTAL PROPELLANT REQUIREMENT
492      C
493      C
494      WPTOT(2) = WDH + WTGH + WVM
495      WHTOT = WDH + WTGH + WVM
496      WPTOT(1) = WDO + WTGO
497      WOTOT = WDO + WTGO
498      C
499      C      OUTPUT APU SUPERCRITICAL DATA
500      CALL OAPUSP (KCYCLE)
501      C
502      C
503      C
504      C *****
505      C COMPUTE WEIGHT AND CHARACTERISTICS OF ALL HEAT EXCHANGERS
506      C *****
507      C
508      JX = 0
509      C
510      C COMPUTE WEIGHT AND CHARACTERISTICS OF THE O2 HEAT EXCHANGER
511      C BETWEEN THE O2 ACCUMULATOR AND THE APU
512      C
513      JX = JX + 1
514      JXH = JX
515      IGAS = 1
516      C
517      WDO MAX = 0.0
518      DO 2010 I = 1,KCYCLE
519      IF(WDRO(I).LT.WDO MAX) GO TO 2010
520      WDO MAX = WDRO(I)
521      I MAX = I

```

***** APUSUP *****

2010 CONTINUE

```

522 C
523   WDOTX(JX,IGAS) = 0.0166 * WDHMAX
524   UCODE(JX,IGAS) = HXCODE(JX,IGAS)
525   HEXHIT(JX,IGAS) = TE(IMAX)
526   HEXCIT(JX,IGAS) = TAO
527   HEXHOT(JX,IGAS) = TD
528   HEXCOT(JX,IGAS) = TPF
529   HEXCOP(JX,IGAS) = PGG
530   HXMRAT(JX,IGAS) = FMR
531
532 C
533   CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS),
534 1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
535 2,HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOXH(JX,IGAS),
536 3,WHXTOT(JX,IGAS))
537
538 C
539   CALL GASGEN(JX,IGAS)
540
541 C
542   COMPUTE WEIGHT AND CHARACTERISTICS OF THE H2 HEAT EXCHANGER
543   BETWEEN THE H2 ACCUMULATOR AND THE APU
544
545 C
546   IGAS = 2
547
548 C
549   WDHMAX = 0.0
550   DO 2020 I = 1,KCYCLE
551   IF(WDRH(I).LT.WDHMAX) GO TO 2020
552   WDHMAX = WDRH(I)
553   IMAX = I
554
555 2020 CONTINUE

```

```

551 C
552   WDOTX(JX,IGAS) = 0.0166 * WDHMAX
553   UCODE(JX,IGAS) = HXCODE(JX,IGAS)
554   HEXHIT(JX,IGAS) = TE(IMAX)
555   HEXCIT(JX,IGAS) = TAO
556   HEXHOT(JX,IGAS) = TD
557   HEXCOT(JX,IGAS) = TPF
558   HEXCOP(JX,IGAS) = PGG
559   HXMRAT(JX,IGAS) = FMR
560
561 C
562   CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS),
563 1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
564 2,HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOXH(JX,IGAS),
565 3,WHXTOT(JX,IGAS))
566
567 C
568   CALL GASGEN(JX,IGAS)
569
570 C
571   COMPUTE WEIGHT AND CHARACTERISTICS OF THE H2 SUPPLEMENTARY GAS
572   HEAT EXCHANGER
573
574 C
575   JX = JX + 1
576   JMX = JX
577   IGAS = 2
578
579 C
580   CALL CSUBPI(TG,FMRG,CPGI)
581   KB = CPGI*(TG-TD)
582   WGHCHX = 0.0
583   DO 2050 I = 1,KCYCLE
584   IF(WGHC(I).LT.WGHCHX) GO TO 2050

```



```

***** APUSUP *****
580      WGHCHX = WGHCH(1)
581      IMAX = 1
582      2050 CONTINUE
583      C
584      QDTSMX = (WGHCHX+WGOC(IMAX)) * K8
585      C
586      TSINMN = TSIN(1)
587      DO 2058 I = 2,KCYCLE
588      IF(TSIN(I).EQ.0.0) GO TO 2058
589      TSINMN = AMIN1(TSINMN,TSIN(I))
590      2058 CONTINUE
591      C
592      WTHMAX = 0.0
593      DO 2055 I = 1,KCYCLE
594      IF(WTH(I).LT.WTHMAX) GO TO 2055
595      WTHMAX = WTH(I)
596      IMAX = I
597      2055 CONTINUE
598      C
599      WDOTX(JX,IGAS) = 0.0166 * WTHMAX
600      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
601      HEXHIT(JX,IGAS) = TG
602      HEXCIT(JX,IGAS) = TSINMN
603      HEXHOT(JX,IGAS) = TD
604      HEXCOT(JX,IGAS) = TAH
605      HEXCOP(JX,IGAS) = APRES(2)
606      HEXCIP(JX,IGAS) = PCH - 25.0
607      HXMRAT(JX,IGAS) = FMRG
608      C
609      IF(HEXCIT(JX,IGAS).EQ.HEXCOT(JX,IGAS)) GO TO 2025
610      C
611      CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS),
612      1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
613      2,HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOXH(JX,IGAS),
614      3,WHXTOT(JX,IGAS))
615      C
616      CALL GASGEN(JX,IGAS)
617      C
618      GO TO 2026
619      C
620      2025 CONTINUE
621      C
622      WRITE (107,885) HXCODE(JX,IGAS)
623      885 FORMAT(/T4,' *** HEAT EXCHANGER - 1,A6, 1 IS NOT IN USE IN THIS PR
624      IOBLEM ***')
625      C
626      2026 CONTINUE
627      C
628      C COMPUTE WEIGHT AND CHARACTERISTICS OF THE O2 HEAT EXCHANGER
629      C BETWEEN THE O2 STORAGE TANK AND THE ACCUMULATOR
630      C
631      JX = JX + 1
632      JHX = JX
633      IGAS = 1
634      C
635      WTMAX = 0.0
636      DO 2030 I = 1,KCYCLE
637      IF(WTO(I).LT.WTMAX) GO TO 2030

```

```

***** APUSUP *****
638      WTHMAX = WTH(I)
639      IMAX = I
640      CONTINUE
641      C
642      WDOTX(JX,IGAS) = 0.0166 * WTHMAX
643      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
644      HEXHIT(JX,IGAS) = TE(IMAX)
645      HEXCIT(JX,IGAS) = TTO(IMAX)
646      HEXHOT(JX,IGAS) = TD
647      HEXCOT(JX,IGAS) = TAO
648      HEXCOP(JX,IGAS) = PCO - HXCDLP(JX,IGAS)
649      HEXCIP(JX,IGAS) = PCO
650      HXMRAT(JX,IGAS) = FHR
651      C
652      CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS),
653      1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
654      2,HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOXH(JX,IGAS),
655      3,WHTOT(JX,IGAS))
656      C
657      CALL GASGEN(JX,IGAS)
658      C
659      C COMPUTE WEIGHT AND CHARACTERISTICS OF THE H2 HEAT EXCHANGER
660      C BETWEEN THE H2 STORAGE TANK AND THE ACCUMULATOR
661      C
662      IGAS = 2
663      C
664      WTHMAX = 0.0
665      DO 2040 I = 1,KCYCLE
666      IF(WTH(I).LT.WTHMAX) GO TO 2040
667      WTHMAX = WTH(I)
668      IMAX = I
669      2040 CONTINUE
670      C
671      WDOTX(JX,IGAS) = 0.0166 * WTHMAX
672      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
673      HEXHIT(JX,IGAS) = TE(IMAX)
674      HEXCIT(JX,IGAS) = TTH(IMAX)
675      HEXHOT(JX,IGAS) = TD
676      IF(DELM(IMAX).EQ.0.0) TSINMN = TAM
677      HEXCOT(JX,IGAS) = TSINMN
678      HEXCOP(JX,IGAS) = PCH - HXCDLP(JX,IGAS)
679      HEXCIP(JX,IGAS) = PCH
680      HXMRAT(JX,IGAS) = FHR
681      C
682      CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS),
683      1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
684      2,HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOXH(JX,IGAS),
685      3,WHTOT(JX,IGAS))
686      C
687      CALL GASGEN(JX,IGAS)
688      C
689      C COMPUTE WEIGHT AND CHARACTERISTICS OF THE O2 HEAT EXCHANGER
690      C OF THE O2 TANK CONDITIONING HEAT EXCHANGER
691      C
692      JX = JX + 1
693      JHX = JX
694      IGAS = 1
695      C

```

```

*****  APUSUP  *****
696      WDOTX(JX,IGAS) = 0.0166 * WDTCP0
697      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
698      HEXHIT(JX,IGAS) = TE(IMAX)
699      HEXCIT(JX,IGAS) = TTO(IMAX)
700      HEXHOT(JX,IGAS) = TD
701      HEXCOT(JX,IGAS) = TD - 100.0
702      HEXCIP(JX,IGAS) = PCO + DELPCP
703      HEXCOP(JX,IGAS) = PCO
704      HXMRAT(JX,IGAS) = FMR
705      C
706      CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS)
707      1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
708      2,HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOOTH(JX,IGAS),
709      3,WHXTOT(JX,IGAS))
710      C
711      CALL GASGEN(JX,IGAS)
712      C
713      COMPUTE WEIGHT AND CHARACTERISTICS OF THE H2 HEAT EXCHANGER
714      OF THE H2 TANK CONDITIONING HEAT EXCHANGER
715      C
716      IGAS = 2
717      C
718      WDOTX(JX,IGAS) = 0.0166 * WDTCPH
719      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
720      HEXHIT(JX,IGAS) = TE(IMAX)
721      HEXCIT(JX,IGAS) = TTH(IMAX)
722      HEXHOT(JX,IGAS) = TD
723      HEXCOT(JX,IGAS) = TD - 100.0
724      HEXCIP(JX,IGAS) = PCH + DELPCP
725      HEXCOP(JX,IGAS) = PCH
726      HXMRAT(JX,IGAS) = FMR
727      C
728      CALL HEATEX(IGAS,JX,WDOTX(JX,IGAS),HEXHIT(JX,IGAS),HEXCIT(JX,IGAS)
729      1,HEXHOT(JX,IGAS),HEXCOT(JX,IGAS),HEXHIP(JX,IGAS),HEXCIP(JX,IGAS),
730      2,HEXHOP(JX,IGAS),HEXCOP(JX,IGAS),HXMRAT(JX,IGAS),WDOOTH(JX,IGAS),
731      3,WHXTOT(JX,IGAS))
732      C
733      CALL GASGEN(JX,IGAS)
734      C
735      CALL OTPHEX
736      C
737      C *****
738      C
739      WRITE (IOT,996)
740      996 FORMAT('////T30,***** THE APU SUPERCRITICAL CALCULATIONS HAVE RE
741      IEN COMPLETED *****')
742      C
743      C *****
744      C
745      C END OF SUPERCRITICAL COMPUTATIONS.
746      C
747      RETURN
748      C
749      END

```

SUBROUTINE BETAB

```
1      SUBROUTINE BETAB(TB,DB,NGAS,VEXB)
2      C
3      C      CALCULATES VOLUME EXPANSIVITY FROM EQUATION OF STATE IN BRITISH
4      C      UNITS
5      C
6      T = TB
7      D = DB
8      N = 1
9      C
10     IF(NGAS.EQ.1) KF = 1
11     IF(NGAS.EQ.18) KF = 2
12     C
13     KF = 1 :CALL IN OXYGEN PROPERTIES (DATA02)
14     KF = 2 :CALL IN NITROGEN PROPERTIES (DATAN2)
15     C
16     IF(KF.EQ.1) CALL DATA02
17     IF(KF.EQ.2) CALL DATAN2
18     C
19     D1 = DPDOB(T,D)
20     D2 = DPDTB(T,D)
21     C
22     VEXB = (1.0/D) * (D2/D1)
23     C
24     RETURN
25     END
```

***** PROCEDURE DEFINITION PROCESSOR - CCACUM

```

1  CACCUM* PROC
2  C
3  PARAMETER NA2=2,KAC2=NA2+1,KAC3=2*NA2+1,KAC4=3*NA2+1,KAC5=4*NA2+1,
4  KAC6=5*NA2+1,KAC7=6*NA2+1,KAC8=7*NA2+1,KAC9=8*NA2+1,KAC0=9*NA2
5  C
6  INTEGER AITYPE,AMTYPE
7  C
8  COMMON /CACCUM/ NAOP (NA2),AMTYPE(NA2),AITYPE(NA2),ADIAM (NA2),
9  AA (NA2),AVOL (NA2),ACHT (NA2),AITHIK(NA2),ACINT (NA2),
10 AHFLUX(NA2),ANBAR (NA2),ANDELP(NA2),APRES (NA2),ATEMP (NA2),
11 WGRACC(NA2),INDXAC(NA2),ACYMT (NA2)
12 C
13 DIMENSION EGAC(KAC0)
14 C
15 EQUIVALENCE (EGAC ,NAOP ),(EGAC(KAC2),AMTYPE),
16 1 (EGAC(KAC3),AITYPE),(EGAC(KAC4),ADIAM ),(EGAC(KAC5),AA ),
17 2 (EGAC(KAC6),AVOL ),(EGAC(KAC7),ACHT ),(EGAC(KAC8),AITHIK),
18 3 (EGAC(KAC9),ACINT )
19 C
20 C
21 C ***** CACCUM VARIABLE DEFINITION
22 C
23 C * AMTYPE - ACCUMULATOR MATERIAL TYPE (SEE CHATRL).
24 C
25 C * AITYPE - ACCUMULATOR INSULATION TYPE (SEE CINSUL).
26 C
27 C * ADIM - DIMENSIONS OF THE ACCUMULATOR
28 C
29 C * AIPRES - ACCUMULATOR INITIAL PRESSURE.
30 C
31 C * AITEMP - ACCUMULATOR INITIAL TEMPRATURE.
32 C
33 C * AHFLUX - ACCUMULATOR HEAT FLUX.
34 C
35 C * AOPRES - ACCUMULATOR OPERATING PRESSURE.
36 C
37 C * AVPRES - ACCUMULATOR VENTING PRESSURE.
38 C
39 C * ANDELP - ACCUMULATOR NOMINAL OPERATING DELTA PRESSURE.
40 C
41 C * AITHIK - ACCUMULATOR INSULATION THICKNESS.
42 C
43 C * ANBAR - NUMBER OF LAYERS OF INSULATION ON ACCUMULATOR
44 C
45 C * NOTE --- EACH OF THE ABOVE VARIABLES ARE SPECIFIED
46 C FOR OXYGEN IN THE FIRST WORD AND HYDROGEN
47 C IN THE SECOND WORD.
48 C *****
49 C
50 END

```

PROCEDURE DEFINITION PROCESSOR - CAPU

```

1 CAPU* PROC
2 C
3   PARAMETER LAPU = 20
4 C
5   REAL KK, MRGGCH, MRGGCO
6 C
7   COMMON /CIAPU/ M ,NAPU ,DELPCP,FMR ,FMRG ,HPR ,MRGGCH,
8 1 MRGGCO,PFH ,PFO ,PGG ,RRFP ,TD ,TDGGH ,TDGGO ,TENV ,
9 2 TFM ,TFO ,TG ,TIT ,TME ,TVH ,TVO ,EQAP1(5),
10 3 EQAP2(11,2),EQAP3(11,2),LAPU1(4,8),LAPU2(4,11),LAPU3(4,11)
11 C
12   COMMON /CVAPU/KK(LAPU), RR(LAPU), WD(LAPU), D (LAPU), TE(LAPU),
13 1 CPE(LAPU),TTH(LAPU),TTO(LAPU),WDA(LAPU),WDB(LAPU),
14 2 WDC(LAPU),WDD(LAPU),WDE(LAPU),WDF(LAPU),WDG(LAPU),
15 3 WDJ(LAPU),WGH(LAPU),WTH(LAPU),WTO(LAPU),HTH(LAPU),
16 4 HTO(LAPU),
17 5 WDRH (LAPU),WDRO (LAPU), PCTHP (LAPU), WGGH (LAPU)
18 6 ,WGGO (LAPU),WDFC (LAPU), WGHC (LAPU), WGO (LAPU)
19 7 ,WDRG (LAPU),WDCC (LAPU), WDEC (LAPU), DELH (LAPU)
20 8 ,DWDH (LAPU),DQDWH(LAPU), DQDWO(LAPU), Q1HDO(LAPU)
21 9 ,Q1ODOT(LAPU),Q2HDO(LAPU), Q2ODOT(LAPU), Q3HDO(LAPU)
22 T ,Q3ODOT(LAPU),TTH2HD(LAPU), TTO2HD(LAPU), PCH2HD(LAPU)
23 1 ,PCO2HD(LAPU),Q2HDT(LAPU), Q2ODTC(LAPU), Q3HDT(LAPU)
24 2 ,Q3ODTC(LAPU),Q4HDO(LAPU), Q5HDO(LAPU), Q6ODOT(LAPU)
25 3 ,Q7ODOT(LAPU),RHOC2(LAPU), RHOC2(LAPU), DQDWH(LAPU)
26 4 ,DQDWO(LAPU),TSIN (LAPU), CPH (LAPU)
27 5 ,CSUBVO(LAPU),CSUBVH(LAPU), PHIO2 (LAPU), PHIH2 (LAPU)
28 6 ,QINTKO(LAPU),QINTKH(LAPU), CPBH (LAPU), WSH (LAPU)
29 C
30   EQUIVALENCE (EQAP1 ,TFF ),(EQAP1(2),WDO ),
31 1 (EQAP1(3),WDH ),(EQAP1(4),WOT ),(EQAP1(5),WTRURN),
32 2,(EQAP2 ,WOCOMP),(EQAP2( 1,2),WHCOMP),(EQAP2( 2,1),WTGO ),
33 3 (EQAP2( 2,2),WTGH ),(EQAP2( 3,1),VTO ),(EQAP2( 3,2),VTH ),
34 4 (EQAP2( 4,1),ATO ),(EQAP2( 4,2),ATH ),(EQAP2( 5,1),WGTOT ),
35 5 (EQAP2( 5,2),WGTHT ),(EQAP2( 6,1),WVHO2 ),(EQAP2( 6,2),WVHM2 ),
36 6 (EQAP2( 7,1),WRO ),(EQAP2( 7,2),WRH ),(EQAP2( 8,1),WAO ),
37 7 (EQAP2( 8,2),WAM ),(EQAP2( 9,1),WSO ),(EQAP2( 9,2),WSH ),
38 8 (EQAP2(10,1),WRAO ),(EQAP2(10,2),WRAH ),(EQAP2(11,1),WOTOT ),
39 9 (EQAP2(11,2),WHTOT )
40 C
41   EQUIVALENCE (EQAP3 ,TGGCO ),(EQAP3( 1,2),TGGCH ),
42 1 (EQAP3( 2,1),WTGGO ),(EQAP3( 2,2),WTGGH ),(EQAP3( 3,1),AREATO ),
43 2 (EQAP3( 3,2),AREATH),(EQAP3( 4,1),VSTO ),(EQAP3( 4,2),VSTH ),
44 3 (EQAP3( 5,1),QTHDOT),(EQAP3( 5,2),QTHDOT),(EQAP3( 6,1),WSVHO ),
45 4 (EQAP3( 6,2),WSVHH ),(EQAP3( 7,2),WSVH ),
46 5 (EQAP3( 8,1),WPTTO ),(EQAP3( 8,2),WPTTH ),(EQAP3( 9,1),WSRO ),
47 6 (EQAP3( 9,2),WSRH ),(EQAP3(10,1),WSOB ),(EQAP3(10,2),WSHB ),
48 7 (EQAP3(11,1),WRSAO ),(EQAP3(11,2),WRSAM )
49 C
50 C
51 C
52 C *****
53 C * NAPU - NUMBER OF APU UNITS
54 C * HPR - HORSEPOWER RATING EACH APU
55 C * FMR - APU TURBINE MIXTURE RATIO
56 C * PGG - APU GAS GEN. INLET GAS PRESSURE
57 C * TIT - TURBINE INLET TEMP
58 C * TD - EXHAUST TEMP FOR HEAT EXCHANGER

```

CAPU

58	C	* FHRG	- FUEL MIXTURE-RATIO FOR SUPPLEMENTAL GAS GEN.
59	C	* PFH	- FINAL H2 TANK PRESSURE
60	C	* PFO	- FINAL O2 TANK PRESSURE
61	C	* TFH	- FINAL H2 TANK TEMPERATURE
62	C	* TFO	- FINAL O2 TANK TEMPERATURE
63	C	* TG	- TEMP. OF EXHAUST PRODUCTS - SUPPL. GAS GEN.
64	C	* DELCP	- DELTA-P OF TANK CIRCULATING PUMP
65	C	* KK	- COR.FACTOR-REF.PROP.FLOW RATE FOR 2060 TIT.
66	C	* RR	- REF.PROP.FLOW RATE (LBS/MIN)
67	C	* WD	- DELIVERED FLOW RATE OVER INTERVAL (I) (LBS/MIN)
68	C	* D	- HEAT OF COMBUSTION PRODUCTS (BTU/LB)
69	C	* TE	- APU TURBINE EXHAUST TEMP.
70	C	* CPE	- EXHAUST SPEC.HT. AT (TE AND FMR)
71	C	* TTH	- TEMP. OF H2 IN TANK DURING INTERVAL(I)
72	C	* TTO	- TEMP. OF O2 IN TANK DURING INTERVAL(I)
73	C	* WDA	- APU EXHAUST FLOW THRU H2-HEX (ACCUM TO GAS GEN)
74	C	* WDB	- APU EXHAUST FLOW THRU H2-HEX (TANK TO ACCUM)
75	C	* WDC	- APU EXHAUST FLOW THRU H2-TANK HEX
76	C	* WDD	- APU EXHAUST FLOW THRU O2-HEX (ACCUM TO GAS GEN)
77	C	* WDE	- APU EXHAUST FLOW THRU O2-HEX (TANK TO ACCUM)
78	C	* WDF	- APU EXHAUST FLOW THRU O2-TANK HEX
79	C	* WDG	- APU EXHAUST FLOW THRU H2-HEX (SURC.ACUM-GAS GEN)
80	C	* WDJ	- APU EXHAUST FLOW THRU O2-HEX (SURC.ACUM-GAS GEN)
81	C	* WGH	- REF.H2 FLOWRATE TO SUPPLEMENTAL GAS GEN.
82	C	* WTH	- H2 FLOW TO ACCUM DURING INTERVAL (I)
83	C	* WTO	- O2 FLOW TO ACCUM DURING INTERVAL (I)
84	C	* HTH	- ENTHALPY OF H2 IN TANK AT INTERVAL(I)
85	C	* HTO	- ENTHALPY OF O2 IN TANK AT INTERVAL(I)
86	C	* WDRH	- WGT.RATE OF H2 FLOWING-INTERVAL(I), (LBS/MIN).
87	C	* WDRO	- WGT.RATE OF O2 FLOWING-INTERVAL(I), (LBS/MIN)
88	C	* PCTHP	- PERCENT HORSEPOWER RECD.-INTERVAL(I)
89	C	* WGGH	- H2 FLOW RATE TO COND.GAS GEN.-INTERVAL(I)
90	C	* WGGO	- O2 FLOW RATE TO COND.GAS GEN.-INTERVAL(I)
91	C	* WDFC	- CORRECTED WDF(I) FOR FLOW RATE CHANGE
92	C	* WGHC	- REF.H2 FLOW TO SUPPLEMENTAL GAS GEN.
93	C	* WGOC	- REF.O2 FLOW TO SUPPLEMENTAL GAS GEN.
94	C	* WDRC	- CORRECTED VALUES OF HEX EXHAUST REQTS.(WDB(I))
95	C	* WDCC	- CORRECTED VALUES OF HEX EXHAUST REQTS.(WDC(I))
96	C	* WDEC	- CORRECTED VALUES OF HEX EXHAUST REQTS.(WDE(I))
97	C	* DELH	- TOTAL ENTHALPY INCREMENT FROM SUPPL.GAS GEN.
98	C	* DHDB	- REF.REDUCTION IN WDR FOR APU EXHAUST AVAIL.
99	C	* TSIN	- T-COLD-IN FOR SUPPLEMENTARY GAS GENERATOR
100	C	* CPH	- SPECIFIC HEAT OF COLD FLUID INTO SUP.GAS GEN.
101	C	* Q00DWH	- HT,XFER.INTO O2 TANK DURING INTERVAL(I)
102	C	* Q00DWO	- HT,XFER.INTO H2 TANK DURING INTERVAL(I)
103	C	* Q1HDOT	- HT,XFER.INTO HEX BETWEEN H2 ACCUM -APU GAS GEN.
104	C	* Q10DOT	- HT,XFER.INTO HEX BETWEEN O2 ACCUM -APU GAS GEN.
105	C	* Q2HDOT	- HT,XFER.INTO HEX BETWEEN H2 TANK - H2 ACCUM.
106	C	* Q20DOT	- HT,XFER.INTO HEX BETWEEN O2 TANK - O2 ACCUM.
107	C	* Q3HDOT	- HT,XFER.INTO H2 TANK - INTERVAL(I)
108	C	* Q30DOT	- HT,XFER.INTO O2 TANK - INTERVAL(I)
109	C	* Q2HDTc	- CORRECTED VALUE OF Q2HDOT DUE TO SUPPL.GAS GEN.
110	C	* Q20DTc	- CORRECTED VALUE OF Q20DOT DUE TO SUPPL.GAS GEN.
111	C	* Q3HDTc	- CORRECTED VALUE OF Q3HDOT DUE TO SUPPL.GAS GEN.
112	C	* Q30DTc	- CORRECTED VALUE OF Q30DOT DUE TO SUPPL.GAS GEN.
113	C	* Q4HDOT	- HT,XFER.INTO HEX BETWEEN H2 ACCUM -APU GAS GEN.
114	C	* Q5HDOT	- HT,XFER.INTO HEX BETWEEN H2 PUMP - H2 ACCUM.
115	C	* Q60DOT	- HT,XFER.INTO HEX BETWEEN O2 ACCUM -APU GAS GEN.

```

***** CAPU *****
116 C      * Q70DOT - HT.XFER.INTO HEX BETWEEN O2 PUMP - O2 ACCUM.
117 C      * TTH2WD - TOTAL USABLE H2 WITHDRAWN TO END INTERVAL(I)
118 C      * TTO2WD - TOTAL USABLE O2 WITHDRAWN TO END INTERVAL(I)
119 C      * PCH2WD - PERCENT USABLE H2 WITHDRAWN TO END INTERVAL(I)
120 C      * PCO2WD - PERCENT USABLE O2 WITHDRAWN TO END INTERVAL(I)
121 C      * RHOCH2 - DENSITY OF H2 IN STORAGE TANK
122 C      * RHOCO2 - DENSITY OF O2 IN STORAGE TANK
123 C      * DQDWH - DQ/DW * WTH(I)
124 C      * DQDWT0 - DQ/DW * WTO(I)
125 C      *****
126 C
127      END

```


***** PROCEDURE DEFINITION PROCESSOR - CCNFIG

```

1 CCNFIG* PROC
2 C
3   PARAMETER ICNF=100
4 C
5   INTEGER CFUNCT,CFTYPE,CHTYPE,CITYPE,CNOPER,CNSTBY
6 C
7   REAL LOD, ITHICK, MACH, NBAR
8 C
9   COMMON /CCNFIG/ ICNFIG(6),CONFIG(ICNF,7),PRES(ICNF),TEMP(ICNF),
10  1 WDOTN(ICNF),MACH(ICNF),MFLG(ICNF),WEIGHT(ICNF),WI(ICNF)
11  2 ,ISTRT(2),KENDC(2),LCNFI(4,6),WTOFSY(5)
12 C
13  DIMENSION FRCOEF(ICNF), LOD(ICNF), DIAM(ICNF), ITHICK(ICNF),
14  1 NBAR (ICNF), CODE(ICNF)
15 C
16  EQUIVALENCE (FRCOEF(1),CONFIG(1,2)),(LOD (1),CONFIG(1,3)),
17  1 (DIAM (1),CONFIG(1,4)),(ITHICK(1),CONFIG(1,5)),
18  2 (NBAR (1),CONFIG(1,6)),(CODE (1),CONFIG(1,7))
19 C
20  EQUIVALENCE (CFUNCT,ICNFIG(1)),(CFTYPE,ICNFIG(2)),
21  1 (CHTYPE,ICNFIG(3)),(CITYPE,ICNFIG(4)),(CNOPER,ICNFIG(5)),
22  2 (CNSTBY,ICNFIG(6))
23 C
24  EQUIVALENCE (ISTRT,IOSTT),(ISTRT(2),IHSTT),(KENDC,KOEND),
25  2 (KENDC(2),KHEND)
26 C
27  EQUIVALENCE (WTOFSY ,O2SWT),(WTOFSY(2),O2IWT),
28  1 (WTOFSY(3),H2SWT),(WTOFSY(4),H2IWT),(WTOFSY(5),TTLSWT)
29 C
30 C
31 ***** CCNFIG VARIABLE DEFINITIONS.
32 C
33  * CONFIGURATION FUNCTION CODE AND TYPE.
34  * CFUNCT = 1, GAS CFTYPE = 1=OXYGEN 2=HYDROGEN
35  *
36  * = 2, ENGINE 1=HI-PRES. 2=LO-PRES.
37  *
38  * = 3, LINE CFTYPE = 10 A FIXED NUMBER
39  *
40  * = 4, CONTROL USES TWO DIGIT INDEX AS FOLLOWS.
41  * IDV = TENS DIGIT (10,20,ETC.)
42  * CFTYPE = UNITS DIGIT (1,2,ETC.)
43  * IDV = 10 FOR LIGHT WGT.CONTROL
44  * = 20 FOR MED. WGT.CONTROL
45  * = 30 FOR HEAVY WGT.CONTROL
46  * = 40 FOR EXT.HEAVY CONTROL
47  * CFTYPE = 1 FOR VALVE
48  * = 2 FOR REGULATOR
49  * = 3 FOR ORIFICE
50  * = 4 FOR FLOW METER
51  *
52  * = 5, FITTING USES TWO DIGIT INDEX AS FOLLOWS.
53  * LDV = TENS DIGIT (10,20,ETC.)
54  * CFTYPE = UNITS DIGIT (1,2,ETC.)
55  * LDV = 10 FOR USE IN LINE ONLY
56  * 20 FOR 4-WAY TEE
57  * 30 FOR 3-WAY TEE

```

***** CCNF10 *****

```

48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C
59 C
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C
104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C

```

40 FOR 90 DEG. ELBOW
 50 FOR 45 DEG. ELBOW
 CFTYPE = 1 FOR TEE
 = 2 FOR ELBOW
 = 6, TAP USES TWO DIGIT INDEX AS FOLLOWS,
 LDV = TENS DIGIT (10,20,ETC.)
 CFTYPE = UNITS DIGIT (1,2,ETC.)
 LDV = 10 FOR USE IN LINE ONLY
 20 FOR 4-WAY TEE
 30 FOR 3-WAY TEE
 CFTYPE = 1 FOR TEE
 = 7, ACCUM NO OPTIONS
 = 8, TANK (SEE TANK ROUTINE)
 = 9, PUMP USES TWO DIGIT INDEX AS FOLLOWS,
 JOPTN = TENS DIGIT (10,20,ETC.)
 CFTYPE = UNITS DIGIT (1,2,ETC.)
 JOPTN = 10 FOR MIN. POWER PUMP
 JOPTN = 20 FOR MIN. WGT. PUMP
 CFTYPE = 1 FOR HI-PRESS. PUMP
 CFTYPE = 2 FOR LO-PRESS XFER PUMP
 = 10, HEX 1=HI-PRES. 2=LO-PRES.
 = 11, END
 * CMTYPE -- CONFIGURATION MATERIAL TYPE.
 CMTYPE = 1, 321/347 STAINLESS STEEL
 = 2, 2219-T87 ALUMINUM ALLOY
 = 3, 6061-T6 ALUMINUM ALLOY
 = 4, INCONEL-718 ALLOY
 = 5, TITANIUM T1-6AL-4V ALLOY
 = 6, CRES VACUUM JACKETED LINE
 = 7, 2219 VACUUM JACKETED LINE
 * CITYPE -- CONFIGURATION INSULATION TYPE.
 CITYPE = 1, DBL.ALUM.MYLAR/SILK NET
 2, DBL.GOLD.MYLAR/SILK NET
 3, DBL.ALUM.MYLAR/TISSUE GLASS
 4, CRINK,DBL.ALUM. MYLAR
 5, NRC-2 CRINKLED ALUMINIZED MYLAR
 6, SUPERFLOC
 7, MICROSPHERES (104-135 MICRON)
 8, POLYURETHANE FOAM
 9, FIBERGLASS BATTING (JM)
 * CNOPER = NUMBER OF OPERATIONAL UNITS (CFUNCT)
 * CNSTBY -- NUMBER OF STANDBY UNITS (CFUNCT)
 * CONFIG -- CONFIGURATION TABLE
 COLUMN 1 CONTAINS THE ABOVE 6 VARIABLES PACKED
 ONE PER BYTE IN THE ORDER THEY ARE
 LISTED FROM LEFT TO RIGHT IN THE WORD.
 COLUMN 2 CONTAINS THE FLOW FRICTION COEFFICIENT.

***** CCONFIG *****

116 C
 117 C
 118 C
 119 C
 120 C
 121 C
 122 C
 123 C
 124 C
 125 END

 * COLUMN 1 CONTAINS THE LENGTH OF A LINE OR THE
 * EFFECTIVE L/D FOR OTHER COMPONENTS.
 * COLUMN 4 CONTAINS THE DIAMETER OF A LINE.
 * COLUMN 5 CONTAINS THE INSULATION THICKNESS FOR
 * A LINE.
 * PRES - PRESSURE AT EACH POINT IN THE CONFIGURATION.

PROCEDURE DEFINITION PROCESSOR - CCNTRL

```

1 CCNTRL* PROC
2 C
3   PARAMETER NBRSR=9,NBRSY=5
4 C
5   INTEGER SCRIT,SYSNUM
6 C
7   COMMON /CCNTRL/ INBLK(NBRSY,5,2),NAMSYS(NBRSY),SCRIT,SYSNUM
8   I ,INTGSY,MDTRC(11),KSUBC(NBRSY,NBRSR),LREPT,JAPUS(2,2)
9 C
10      INBLK = CONTROLS INPUT SELECTION IN COMPIL
11 C      SCRIT = 1 FOR SUB-CRITICAL
12 C      = 2 FOR SUPER-CRITICAL
13 C      SYSNUM = 1 ACPs
14 C      = 2 APU
15 C      = 3 EC/LSS
16 C      = 4 FUEL CELL
17 C      = 5 OHS
18 C
19 CARD COL, MDTRC( ) = DIAGNOSTIC TRACE SWITCH FOR CRYCON (OFF=0)
20 C      (70) (1) = 1 TURN ON ACCRES
21 C      (71) (2) = 1 TURN ON ACQWT
22 C      (72) (3) = 1 TURN ON APUSUB OR APUSUP
23 C      (73) (4) = 1 TURN ON CMPCAL
24 C      (74) (5) = 1 TURN ON FUELCL
25 C      (75) (6) = 1 TURN ON CONSUM
26 C      (76) (7) = 1 TURN ON ECLSS
27 C      (77) (8) = 1 TURN ON LIGRES
28 C      (78) (9) = 1 TURN ON TANK
29 C      (79) (10) = 1 TURN ON TSIZE1
30 C      (80) (11) = 1 TURN ON WTACC
31 C
32 C      MDTRC(1) IS CARD COL 70,--- MDTRC(11) IS CARD COL 80
33 C      OF THE SYSTEM SPECIFICATION CARD
34 C
35 C
36 END

```

***** PROCEDURE DEFINITION PRØCESSØR - CDCYCL

```

1 CDCYCL* PROC
2 C
3   PARAMETER  ICDL=30,ICDL1=2*ICDL,ICDL2=ICDL1+1
4 C
5   COMMON /CDCYCL/ DCYCLT, NDCYCL, DCYCLE(ICDL), NEOP(ICDL),
6     1 PSI(ICDL), HP(ICDL), PAMB(ICDL), KCYCLE
7     2 ,PKW(ICDL),RPRTIM(ICDL)
8 C
9   ***** CDCYCL VARIABLE DEFINITION
10 C
11 C   * PSI   - % OF ENGINE DEGRADATION DUE TO MIB (INPUT).
12 C   *
13 C   * NDCYCL - NUMBER OF ELEMENTS IN DCYCLE (INPUT).
14 C   *
15 C   * DCYCLE - ARRAY OF DUTY CYCLE TIMES. ODD SUBSCRIPTED
16 C   *          WORDS CONTAIN BURN TIMES, EVEN SUBSCRIPTED
17 C   *          WORDS CONTAIN COAST TIMES (INPUT).
18 C   *
19 C   * DCYCLT - TOTAL OPERATING TIME FROM DUTY CYCLE.
20 C   *
21 C   * NEOP   - NUMBER OF OPERATING UNITS.
22 C   *
23 C   * HP     - HORSEPOWER LOAD PER OPERATING UNIT.
24 C   *
25 C   * PAMB   - AMBIENT PRESSURE AROUND OPER. UNITS.
26 C   *
27 C   * KCYCLE - INTEGER COUNT OF (ICDL) INPUT VALUES
28 C   *
29 C   * PKW    - POWER IN KILOWATTS
30 C   *
31 C   * RPRTIM - CABIN OR AIRLOCK REPRESSURIZATION TIME
32 C   *
33 C   *****
34 C
35   END

```

PROCEDURE DEFINITION PROCESSOR - CECLSS

CECLSS* PROC

C

PARAMETER LX2=2, LXV=12

C

REAL NIENH

C

REAL N2LCON, N2RES, N2REPR, N2CONS, N2TOTU, N2LWT, N2RHO, N2H,
N2TEMP, N2LRES, LINDIA, NRHO, N2TEM

C

COMMON /CILSS/ MDAYS, NCREW, NRPRES, NDARES, O2FNOM, GLKRAT,
TENVR, CABVOL, O2MCON, O2LCON, N2LCON, O2RES,
N2RES, POPO2, POPH2, PVP02, PVPN2, T02IN,
TN2IN, GASWGT, O2REPR, N2REPR, O2CONS, N2CONS,
HLSO, O2TOTU, N2TOTU, WDTMXO, WDTMXN, TEMPO2,
HLSN, TEMPN2, TKOMXT, TKNMXT, ZFO, ZFN,
WVO2, WVN2, OLKOTK, GLKNTK, WTVJO, WTVJH,
NPHO, O2LRES, N2LRES, HWTOMX, HWTNMX, TWTOHX,
PSET1, PSET2, TWTHMX, TOTWMX, TOTWAT, TOTPOW,
N2TEM

C

COMMON /CVLSS/ TLSNOM(LX2), RHOBEG(LX2), TKFTEM(LX2), TKFPRS(LX2),
TAU (LXV), WTO2 (LXV), WTN2 (LXV), O2MHT (LXV),
O2LWT (LXV), N2LWT (LXV), WDOTON(LXV), WDOTNN(LXV),
WDOTOR(LXV), WDOTNR(LXV), WDT02 (LXV), WDTN2 (LXV),
TKO2DP(LXV), TKN2DP(LXV), PCOXWD(LXV), PCH2WD(LXV),
O2RHO (LXV), N2RHO (LXV), O2TEMP(LXV), N2TEMP(LXV),
DODMO2(LXV), DPDUO2(LXV), DODMN2(LXV), DPDUN2(LXV),
O2H (LXV), N2H (LXV), QDOR (LXV), QDTHR (LXV),
HWATO2(LXV), HWATN2(LXV), QDTTKO(LXV), QDTTKN(LXV),
TWATO2(LXV), TWATN2(LXV), RHOEND(LX2), WTRSID(LX2),
VOLTk (LX2), ARETK (LX2), O02LK (LXV), QN2LK (LXV),
QREGDO(LXV), QREGDN(LXV), WTVHTO(LXV), WTVHTN(LXV),
TOTHTL(LX2), DITK (LX2), DIVJ (LX2), ROFTU (LX2),
WTPV (LX2), HTRFLX(LX2), LINDIA(LX2), PLSNOM(LX2),
HTRDIA(LX2), HTRLNG(LX2), TIM (LXV), TNONOP(LXV)

D

C

* MDAYS - DURATION OF MISSION IN DAYS
* NCREW - NUMBER OF PERSONEL IN CREW
* NRPRES - NUMBER OF AIRLOCK REPRESSURATIONS
* NDARES - NUMBER OF DAYS SUPPLY OF RESERVE O2 AND N2
* O2FNOM - OXYGEN REQUIRED FOR CREW - LBS PER MAN-DAY
* GLKRAT - VEHICLE GAS LEAKAGE RATE - LBS PER DAY (AS AIR)
* TLSNOM - NOMINAL O2 OR N2 DELIVERED TEMPERATURE
* RHOBEG - O2 OR N2 INITIAL LOADED DENSITY
* TKFTEM - O2 OR N2 FINAL TANK TEMPERATURE
* TKFPRS - O2 OR N2 FINAL TANK PRESSURE
* TENVR - LSS ENVIRONMENTAL TEMPERATURE (AMBIENT)
* CABVOL - CABIN OR AIRLOCK VOLUME - CU.FT.

C

END

***** PROCEDURE DEFINITION PROCESSOR - CENG

```

1  CENG*  PROC
2  C
3  C      PARAMETER  NECDL = 30
4  C
5  C      REAL MIXRAT,ISP
6  C
7  C      COMMON /CENG/GITEMP,GIPRES,THRUST,PSUBC,EXPRAT,MIXRAT,NENG,
8  C      1      ISP,WDOT,TIPWT,ENGWT,WDOTI(2),WDOTT(2),
9  C      2      WDOTJ(NECDL,2)
10 C
11 C      ***** CENG VARIABLE DEFINITION
12 C
13 C      * GITEMP = GAS INLET TEMPERATURE (INPUT).
14 C
15 C      * GIPRES = GAS INLET PRESSURE (INPUT).
16 C
17 C      * THRUST = ENGINE THRUST (INPUT).
18 C
19 C      * PSUBC = CHAMBER PRESURE (INPUT).
20 C
21 C      * EXPRAT = EXPANSION RATIO (INPUT).
22 C
23 C      * MIXRAT = MIXTURE RATIO (INPUT).
24 C
25 C      * NENG = NUMBER OF ENGINES.
26 C
27 C      * ISP = SPECIFIC IMPULSE (CALC IN ENGINE).
28 C
29 C      * WDOT = FLOW RATE (CALC IN ENGINE).
30 C
31 C      * TIPWT = TOTAL IMPULSE PROPELLANT WEIGHT (CALC IN ENGINE).
32 C
33 C      * ENGWT = ENGINE WEIGHT (CALC IN ENGINE).
34 C
35 C      * WDOTI = FLOW RATE FOR O2 AND H2 FROM ENGINE TO TAP.
36 C
37 C      * WDOTT = FLOW RATE FOR O2 AND H2 FROM TAP THROUGH REST
38 C      ***** OF SYSTEM.
39 C
40 C      END

```

PROCEDURE DEFINITION PROCESSOR - CFLRAT

```
1 CFLRAT* PROC
2 C
3   COMMON /CFLRAT/ EQRT(6,2),LFRT(3,7)
4 C
5   EQUIVALENCE      (EQRT      ,WFLO5 ),(EQRT(1,2),WFLO7 ),
6   1 (EQRT(2,1),WFLO6 ),(EQRT(2,2),WFLO8 ),(EQRT(3,1),WDTPTO),
7   2 (EQRT(3,2),WDTPTF),(EQRT(4,1),WFLO1 ),(EQRT(4,2),WFLO3 ),
8   3 (EQRT(5,1),WFLO2 ),(EQRT(5,2),WFLO4 ),(EQRT(6,1),WDHXT0),
9   4 (EQRT(6,2),WDHXTF)
10 C
11  END
```


***** PRØCEDURE DEFINITIØN PRØCESSØR - CFUID

```
1 CFLUID* PROC  
2 C  
3 C  
4 END
```

FUNCTION CFTW

```

1      FUNCTION CFTW (D,P,IDV)
2      C
3      REAL K1,K2,K3,K4
4      C
5      DIMENSION K1(4),K2(4),K3(4),K4(4),C1(4),C2(4),C3(4),C4(4)
6      C
7      DATA K1/0.040,0.057,0.073,0.090/
8      DATA K2/0.057,0.073,0.090,0.107/
9      DATA K3/1.000,2.500,3.300,5.500/
10     DATA K4/2.500,3.300,5.500,7.700/
11     DATA C1/1.750,3.950,5.730,8.910/
12     DATA C2/3.950,5.730,8.910,12.35/
13     DATA C3/0.800,1.500,2.500,3.500/
14     DATA C4/1.500,2.500,3.500,4.500/
15     C
16     C      SET IDV TO EXTRA HEAVY IF NOT INPUT
17     IF (IDV.EQ. 0) IDV = 4
18     IF (D.LE.1.) GO TO 10
19     IF (P.GT.300.0.AND.D.GT.3.5) GO TO 5
20     IF (P.GT.1000.0.AND.D.LE.3.5) GO TO 5
21     CFTW = K1(IDV)*D*D*D + C1(IDV)
22     RETURN
23     5 CFTW = K2(IDV)*D*D*D + C2(IDV)
24     RETURN
25     10 IF (P.GE.1000.) GO TO 15
26     CFTW = K3(IDV)*D + C3(IDV)
27     RETURN
28     15 CFTW = K4(IDV)*D + C4(IDV)
29     RETURN
30     END

```

***** PROCEDURE DEFINITION PROCESSOR - CFUEL

```

1 CFUEL* PROC
2 C
3   PARAMETER LFC = 12, LFD = 2
4 C
5   REAL MRFC
6 C
7   COMMON /CIFUEL/ SRCFC, MRFC, POWTOT, WRFORP, WOCONS, WHCONS,
8     1 PKHMAX, QFCTOT, QDTFC, TF21IN, TF21OU, TFOFC,
9     2 TFHFC, PFOFC, PFHFC, QTOTR, QEXCES, WF21MX,
10    3 DGAMIN, TKOMAX, TKHMAX, QMXTKO, QMXTKH, WDTFCO,
11    4 WDTCFH, WOCMP, WHCMP, POWMAX, DELTCP, WRRSRV,
12    5 WORSRV, WRRSRV, QLEAKO, QLEAKH, WVHO, WVHH,
13    6 HOVENT, HHVENT, SPWT1, SPWT2, NFCOP, NFCSTB,
14    7 SPWTEC, FCWGT, PRFCOP, WDTFMX, PLSET1, PLSET2,
15    8 FCVOLT
16 C
17   COMMON /CVFUEL/ WRP (LFC), WDTFCO(LFC), WDTFCH(LFC), WDOTMX(LFD),
18     1 QAVAIL(LFC), WDTF21(LFC), PRFCMN(LFD), TFCNOM(LFD),
19     2 TKO2WD(LFC), TKH2WD(LFC), PCWD02(LFC), PCWDH2(LFC),
20     3 RHOT02(LFC), RHOTH2(LFC), DODWO (LFC), DODWH (LFC),
21     4 TKO (LFC), TKH (LFC), HTKO (LFC), HTKH (LFC),
22     5 Q1ODTR(LFC), Q1HODTR(LFC), WDT1FO(LFC), WDT1FH(LFC),
23     6 Q2ODTR(LFC), Q2HODTR(LFC), WDT2FO(LFC), WDT2FH(LFC),
24     7 CSRVFO(LFC), CSRVFH(LFC), PHIF02(LFC), PHIFH2(LFC),
25     8 QSUNR (LFC), DQANET(LFC), RHOFIL(LFD), RHOFIN(LFD),
26     9 WTRES (LFD), VOLTNK(LFD), AREATK(LFD), QLKO (LFC),
27     T QLKH (LFC), WRTOTL(LFD), DIATK (LFD), DIAVJ (LFD),
28     I WCIRCP(LFD), RHOFU(LFD), WTPVT (LFD), WTVJ (LFD),
29     B WOREP (LFC), WHREP (LFC), VJANUL(LFD), TKHXDI(LFD),
30     C PRGRAT(LFD), PRGTIM(LFD), PRGINT(LFD), PURGAS(LFD)
31 C
32 C *****
33 C *
34 C *
35 C *****
36 C
37 END

```

PROCEDURE DEFINITION PROCESSOR - CHEX

CHEX* PROC

C

PARAMETER MXX=10, MXX2 = 2*MXX+1, MXX3 = 4*MXX+1, MXX4 = 6*MXX+1,
1 MXX5 = 8*MXX+1, MXX6 = 10*MXX+1, MXX7 = 12*MXX+1, MXX8 = 14*MXX+1,
2 MXX9 = 16*MXX+1, MXX10 = 18*MXX+1, MXX11 = 20*MXX+1, MXX12 = 22*MXX,
3 MYH2 = 6*MXX+1, MYH3 = 12*MXX+1, MYH4 = 18*MXX+1, MYH5 = 24*MXX+1,
4 MYH6 = 30*MXX+1, MYH7 = 36*MXX+1, MYH8 = 42*MXX+1, MYH9 = 48*MXX

C

REAL NTUU

C

COMMON /CHEX/ NUMHEX, HEXCIT(MXX,2), HEXCOT(MXX,2), CPCPDF(MXX,2),
1 WDOTCF(MXX,2), HEXHIT(MXX,2), HEXHOT(MXX,2), CPHOTF(MXX,2),
2 WDOTH(MXX,2), EPSLNC(MXX,2), EPSLNH(MXX,2), EPSLNS(MXX,2),
3 HEXHIP(MXX,2), HEXHOP(MXX,2), HEXCIP(MXX,2), HEXCOP(MXX,2),
4 HXHOLP(MXX,2), HXCOLD(MXX,2), HXMRAT(MXX,2), HXASSY(MXX,2),
5 WGGFU(MXX,2), WGGFX(MXX,2), WGGSBT(MXX,2), WHXTOT(MXX,2),
6 IISU(MXX,2), NSSK(MXX,2), NSUK(MXX,2), UCODE(MXX,2),
7 TCRU(MXX,3,2), FWDTHU(MXX,3,2), FDPUC(MXX,3,2),
8 CRU(MXX,3,2), NTUU(MXX,3,2), UAU(MXX,3,2),
9 WOUAU(MXX,3,2), WTHXU(MXX,3,2), UNAM(2,4), WDHX(2),
10 LHX1(4,10), LHX2(4,14), LHX3(5), UOA(MXX,2), DH(MXX,2),
11 A HXCODE(MXX,2), HXLENGTH(MXX,2)

C

DIMENSION UOTHX(MXX), UOTHY(MYH9)

C

EQUIVALENCE (UOTHX,HEXCIT),(UOTHX(MXX2),HEXCOT),
1 (UOTHX(MXX3),CPCPDF),(UOTHX(MXX4),WDOTCF),(UOTHX(MXX5),HEXHIT),
2 (UOTHX(MXX6),HEXHOT),(UOTHX(MXX7),CPHOTF),(UOTHX(MXX8),WDOTH),
3 (UOTHX(MXX9),EPSLNC),(UOTHX(MXX10),EPSLNH),(UOTHX(MXX11),EPSLNS),
4 (UOTHY,TCRU),(UOTHY(MYH2),FWDTHU),(UOTHY(MYH3),FDPUC),
5 (UOTHY(MYH4),CRU),(UOTHY(MYH5),NTUU),(UOTHY(MYH6),UAU),
6 (UOTHY(MYH7),WOUAU),(UOTHY(MYH8),WTHXU)

C

***** HEXHIT - HEX HOT INLET TEMPERATURE (HEX GAS GEN. TC)

C

* HEXHOT - HEX HOT OUTLET TEMPERATURE

C

* HEXCIT - HEX COLD INLET TEMPERATURE

C

* HEXCOT - HEX COLD OUTLET TEMPERATURE

C

* HEXHIP - HEX HOT INLET PRESSURE (HEX GAS GEN. PC)

C

* HEXHOP - HEX HOT OUTLET PRESSURE

C

* HEXCIP - HEX COLD INLET PRESSURE

C

* HEXCOP - HEX COLD OUTLET PRESSURE

C

* HXHOLP - HEX HOT SIDE DELTA-PRESSURE

C

* HXCOLD - HEX COLD SIDE DELTA-PRESSURE

C

* HXMRAT - MIXTURE RATIO (O/F) OF HEX GAS GENERATOR

C

* WDOTH - HEX HOT GAS FLOW RATE

C

*

```

***** CHEX *****
58 C      * WDHX  - FLORAT ESTIMATE OF HEX FLOW READ (EACH FLUID)
59 C      *
60 C      * WHXTOT - HEX UNIT WEIGHT
61 C      *
62 C      * HXASSY - HEX ASSEMBLY WEIGHT (HEX + GAS GEN.)
63 C      *
64 C      * HXCODE - HEX CODE SYMBOL NAME
65 C      *
66 C      * NOTE --- THE ABOVE VARIABLES ARE SPECIED FOR OXYGEN IN
67 C      *      THE FIRST WORD AND HYDROGEN IN THE SECOND
68 C      *      WORD IN THE SECOND SUBSCRIPT.
69 C      *
70 C      *****
71      END

```

PROCEDURE DEFINITION PROCESSOR - CHSØRC

```

1 CHSØRC= PROC
2 C
3   PARAMETER MHSC = 10
4 C
5   INTEGER HSTYPE
6 C
7   COMMON /CHSØRC/ HSTYPE(MHSC,2), HSMRAT(MHSC,2), HSOTEM(MHSC,2),
8   1 HSAEE(MHSC,2), HSPRES(MHSC,2), HSWGHT(MHSC,2), HSGTOT(MHSC,2),
9   2 HSGCPE(MHSC,2), HSOREG(MHSC,2), HFTOT (2), QTREQ (2),
10  3 NUMHSA, LHS1(5,6), LHS2(5,14), ELCPOW(MHSC,2)
11 C
12 C   ***** HSTYPE - TYPE OF HEAT SOURCE
13 C   *
14 C   *           = 1, GAS GENERATOR ONLY
15 C   *           = 2, WASTE HEAT INPUT ONLY
16 C   *           = 3, GAS GEN. AND WASTE HEAT INPUT
17 C   *
18 C   * HSMRAT - HEAT SOURCE MIXTURE RATIO
19 C   *
20 C   * HSOTEM - HEAT SOURCE OUTLET TEMPERATURE
21 C   *
22 C   * HSAEE - HEAT SOURCE AVAILABLE ENERGY
23 C   *
24 C   * HSPRES - HEAT SOURCE OUTLET PRESSURE
25 C   *
26 C   * HSWGHT - HEAT SOURCE WEIGHT
27 C   *
28 C   * HSGTOT - TOTAL HOT FLUID WEIGHT
29 C   *
30 C   * HSGCPE - SPECIFIC HEAT OF HOT FLUID
31 C   *
32 C   * HSOREG - TOTAL HEAT REQUIRED FROM HOT FLUID
33 C   *
34 C   * HFTOT - CUMULATIVE HOT FLUID FOR SYSTEM
35 C   *
36 C   * QTREQ - CUMULATIVE HEAT REQUIRED FOR SYSTEM
37 C   *
38 C   * NUMHSA - NUMBER OF HEAT SOURCE SETS
39 C   *
40 C   * LHS1 - HEAT SOURCE OUTPUT LABELS
41 C   *
42 C   * ELCPOW - ELECTRIC HEAT SOURCE - WATTS
43 C   *
44 C   * IN THE ABOVE VARIABLES THE FIRST WORD OF THE SECOND
45 C   * ARRAY IS FOR OXYGEN, AND THE SECOND WORD IS FOR
46 C   * HYDROGEN.
47 C   *****
48 END

```

***** PROCEDURE DEFINITION PROCESSOR - CHTX

```
1 CHTX* PROC
2 C
3 C      *****
4 C      *
5 C      * THE VARIABLES IN CHTX ARE FOR OUTPUT USE AND ARE
6 C      * DEFINED IN S.R. HEATEX
7 C      *
8 C      * NOTE --- THE ABOVE VARIABLES ARE SPECIED FOR OXYGEN IN
9 C      * THE FIRST WORD AND HYDROGEN IN THE SECOND
10 C      * WORD IN THE SECOND SUBSCRIPT.
11 C      *
12 C      *****
13 END
```

PROCEDURE DEFINITION PROCESSOR - CIOUNT

```

1 CIOUNT* PROC
2 C
3   COMMON /CIOUNT/ IOUNIT(14),IIN,IOT
4 C
5   EQUIVALENCE      (MURD1 ,IOUNIT( 1)),(MURD2 ,IOUNIT( 2)),
6   1(MURD3 ,IOUNIT( 3)),(MURD4 ,IOUNIT( 4)),(MURD5 ,IOUNIT( 5)),
7   2(MURD6 ,IOUNIT( 6)),(NTAPE1,IOUNIT( 7)),(NTAPE2,IOUNIT( 8)),
8   3(NTAPE3,IOUNIT( 9)),(NTAPE4,IOUNIT(10)),(NTAPE5,IOUNIT(11)),
9   4(NTAPE6,IOUNIT(12)),(NTAPE7,IOUNIT(13)),(NTAPE8,IOUNIT(14))
10 C
11 C
12 C   IOUNIT - AND ARRAY OF VALUES WITH ARE THE LOGICAL UNIT NUMBERS
13 C   AS DETAILED IN SEC. 4.4.1 OF THE LMSC DIGITAL
14 C   COMPUTER SYSTEMS MANUAL (LMSC-685147).
15 C
16 END

```


***** PROCEDURE DEFINITION PRØCESSØR - CKEYS

```
1 .CKEYS* PROC
2 C
3     COMMON /CKEYS/ KEY1,KEY2
4 C
5     END
```

PROCEDURE DEFINITION PROCESSOR - CMATRL

CMATRL* PROC

C

REAL MINTHK

C

COMMON/CMATRL/RHOL(10),RHOI(10),RHOIS(10),MINTHK(15)

C

C

**** RHOL(I) IS THE DENSITY OF THE SELECTED LINE MATL. (LB/CU.FT)

C

* I = 1, FOR 321/347 STAINLESS STEEL

C

* I = 2, FOR 2219-T87 ALUM. ALLOY

C

* I = 3, FOR 6061-TL ALUM. ALLOY

C

* I = 4, FOR INCONEL-718

C

* I = 5, FOR TITANIUM TI-6AL-4V ALLOY

C

C

C

**** RHOI(I)-THE DENSITY OF THE CHOSEN INSULATION MATRL.(LB/CU.FT)
* OPTIMIZED LAYER DENSITY VALUES.

C

C

* I = 1, FOR DOUBLE ALUMINIZED MYLAR/SILK NET (50 LAYER/INCH)

C

* I = 2, FOR DOUBLE GOLDIZED MYLAR/SILK NET (50 LAYER/INCH)

C

* I = 3, FOR DOUBLE ALUMINIZED MYLAR/TISSUE GLASS (100-LAY/IN)

C

* I = 4, FOR CRINKLED DBLE.ALUM.MYLAR/TISS.GLASS (30-LAY/IN)

C

* I = 5, FOR NPC-2 CRINKLED ALUM. MYLAR (40 LAYER/INCH)

C

* I = 6, FOR SUPERFLOC (30 LAYER/INCH)

C

* I = 7, FOR MICROSPHERES (104 MICRON DIA.) (PER CUBIC FOOT)

C

* I = 8, FOR POLYURETHANE FOAM (PER CUBIC FOOT)

C

* I = 9, FOR FIBERGLASS (J.M)(800-1200 DEG.R) (PER CUBIC FOOT)

C

* REF.'HDBK FOR THERMAL DESIGN'-LMSC-A847882,VOL.2,25 JUNE 1967
* 'MICROSPHERES'-DATA OF PARMLEY AND CUNNINGHAM, LMSC.

C

C

C

C

**** RHOIS(I)--DENSITY OF THE CHOSEN INSULATION MATRL.(LB/CU.FT)
* ON PER LAYER BASIS FOR SPECIFYING LAYER FENSITY.

C

C

* I = 1, FOR DOUBLE ALUMINIZED MYLAR/SILK NET (PER LAYER/INCH)

C

* I = 2, FOR DOUBLE GOLDIZED MYLAR/SILK NET (PER LAYER/INCH)

C

* I = 3, FOR DOUBLE ALUMINIZED MYLAR/TISSUE GLASS (PER-LAY/IN)

C

* I = 4, FOR CRINKLED DBLE.ALUM.MYLAR/TISS.GLASS (PER-LAY/IN)

C

* I = 5, FOR NPC-2 CRINKLED ALUM. MYLAR (PER LAYER/INCH)

C

* I = 6, FOR SUPERFLOC (PER LAYER/INCH)

C

* I = 7, FOR MICROSPHERES (104 MICRON DIA.) (PER CUBIC FOOT)

C

* I = 8, FOR POLYURETHANE FOAM (PER CUBIC FOOT)

C

* I = 9, FOR FIBERGLASS (J.M)(800-1200 DEG.R) (PER CUBIC FOOT)

C

* REF.'HDBK FOR THERMAL DESIGN'-LMSC-A847882,VOL.2,25 JUNE 1967
* 'MICROSPHERES'-DATA OF PARMLEY AND CUNNINGHAM, LMSC.

C

C

C

C

MINIMUM THICKNESS DATA (INCHES)

C

C

***** .CHATRL *****

58 C
59 END

PROCEDURE DEFINITION PROCESSOR - CMOTOR

```
1  CMOTOR* PROC  
2  C  
3      REAL HEFF,MSS  
4  C  
5      COMMON /CMOTOR/NTYPE,HEFF,MSS,PONSTY,BWEIGHT(2)  
6  C  
7      END
```

SUBROUTINE CMPCAL :*****

639717*TPFS.CMPCAL

```
1 C
2 C
3 C      * * * * *
4 C      * ROUTINE NAME - PRESSURE DROP COMPUTATION *
5 C      * FOR ENTIRE CONFIGURATION *
6 C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
7 C      * PROGRAMMER - R. BOLLINGER 1943 102 26933 *
8 C      * DATE CODED - 3/17/70 *
9 C      * * * * *
10 C      SUBROUTINE CMPCAL
11 C
12 C      INTEGER GSTATE
13 C
14 C      LOGICAL PAGE, JP
15 C
16 C
17 C      REAL NPSPR
18 C
19 C      INCLUDE CACCUH
20 C      INCLUDE CCNFIG
21 C      INCLUDE CCNTRL
22 C      INCLUDE CDCYCL
23 C      INCLUDE CENG
24 C      INCLUDE CHEX
25 C      INCLUDE CFUEL
26 C      INCLUDE CHSIRC
27 C      INCLUDE CIOUNT
28 C      INCLUDE CNAIES
29 C      INCLUDE CMOTOR
30 C      INCLUDE CONST
31 C      INCLUDE CPAGE
32 C      INCLUDE CPUMP
33 C      INCLUDE CTANK
34 C      INCLUDE CTURBH
35 C      INCLUDE TABLOK
36 C
37 C      ***** INITIALIZE THE ROUTINE
38 C
39 C      IDX = 0
40 C      ISIGN = 1
41 C      JKM = 0
42 C      WGGTOT(1) = 0.0
43 C      WGGTOT(2) = 0.
44 C      HFTOT(1) = 0.0
45 C      HFTOT(2) = 0.0
46 C      QTREQ(1) = 0.0
47 C      QTREQ(2) = 0.0
48 C      C1 = 1152.0/(GRAVTV*PI**2)
49 C      IF(PAGE(0)) WRITE (IOT,6050)
50 C      WRITE (IOT,6020)
51 C      JP = PAGE (3)
52 C
53 C      ***** START OF CONFIGURATION PROCESSING LOOP
54 C
55 C      DO 1000 I=1,ICNF
56 C      IDX = IDX + ISIGN
57 C      MACH(IDX) = 0.0
```

```

***** CMPCAL *****
58      MFLG(IDX) = 6H
59      CALL GETCON(IDX)
60      C
61      C      ***** BRANCH TO THE REQUIRED CONFIGURATION TYPE. SEE CCHFIG
62      C      ***** FOR BRANCH DEFINITIONS.
63      C
64      GO TO (100,200,300,400,450,500,450,450,400,405,600,700,800,900,
65      I 230,250,270,1100), CFUNCT
66      C
67      C      ***** SETUP THE GAS TYPE *****
68      C
69      100 IGAS = CFTYPE
70      GSTATE = ICNFIG(5)
71      IF (IGAS.EQ. JKM) GO TO 110
72      JKM = IGAS
73      ISIGN = 1
74      ISTRT(IGAS) = IDX + 1
75      JX = 0
76      110 CONTINUE
77      C
78      IF (IGAS.EQ.2.AND.GSTATE.EQ.1) GO TO 111
79      GO TO 112
80      111 IF (PAGE(0)) WRITE (IOT,6051)
81      WRITE (IOT,6020)
82      JP = PAGE(3)
83      C
84      112 CONTINUE
85      C
86      IF (11.EQ.1) GO TO 999
87      IF (IGAS.EQ.2.AND.GSTATE.EQ.1) GO TO 999
88      PRES(IDX) = PRES(IDX - ISIGN)
89      WDOTN(IDX) = WDOTN(IDX - ISIGN)
90      TEMP(IDX) = TEMP(IDX - ISIGN)
91      GO TO 999
92      C
93      C      ***** PROCESS AN ENGINE *****
94      C
95      200 WDOTN(IDX) = WDOTI(IGAS)
96      PRES(IDX) = GIPRES
97      TEMP(IDX) = GITEMP
98      WEIGHT(IDX) = ENGWT
99      GO TO 999
100     C
101     C      ***** PROCESS AN APU TURBINE UNIT *****
102     C
103     230 WDOTN(IDX) = WDOTI(IGAS)
104     PRES(IDX) = HEXCOP(1,IGAS)
105     TEMP(IDX) = HEXCOT(1,IGAS)
106     GO TO 999
107     C
108     C      ***** PROCESS A FUEL CELL *****
109     C
110     250 WDOTN(IDX) = WDOTI(IGAS)
111     PRES(IDX) = PRFCOP
112     TEMP(IDX) = TFCNOH(IGAS)
113     ISIGN = 1
114     GO TO 999
115     C

```

```

***** CMPCAL *****
116 C
117 C ***** PROCESS A LINE *****
118 C
119 300 FLD = FRCOEF(IDX)*LOD(IDX)/DIAM(IDX)
120 LDV = CFTYPE/10
121 CFTYPE = CFTYPE - LDV * 10
122 310 WDOTH(IDX) = WDOTH(IDX-ISIGN)
123 TEMP(IDX) = TEMP(IDX-ISIGN)
124 GO TO 510
125 C
126 C ***** PROCESS A CONTROL *****
127 C
128 400 FLD = FRCOEF(IDX)*LOD(IDX)
129 IDV = CFTYPE /10
130 CFTYPE = CFTYPE - IDV * 10
131 DIAM(IDX) = AMINI(DIAM(IDX+1),DIAM(IDX-1))
132 GO TO 310
133 C
134 C ***** PROCESS A REGULATOR *****
135 C
136 405 FLD = FRCOEF(IDX)*LOD(IDX)
137 IDV = CFTYPE /10
138 CFTYPE = CFTYPE - IDV * 10
139 DIAM(IDX) = AMINI(DIAM(IDX+1),DIAM(IDX-1))
140 WDOTH(IDX) = WDOTH(IDX-ISIGN)
141 TEMP(IDX) = TEMP(IDX-ISIGN)
142 C
143 IX = IDX - ISIGN
144 IF(APRES(IGAS) .EQ. 0.0) GO TO 406
145 DLPREG = (APRES(IGAS) - ANDELP(IGAS)/2.0) - PRES(IX)
146 PRES(IDX) = PRES(IX) + ISIGN * DLPREG
147 GO TO 561
148 C
149 406 CONTINUE
150 DLPREG = HEXCOP(1,IGAS) - PRES(IX)
151 PRES(IDX) = PRES(IX) + ISIGN * DLPREG
152 GO TO 561
153 C
154 C ***** PROCESS A FITTING *****
155 C
156 450 FLD = FRCOEF(IDX) * LOD(IDX)
157 LDV = CFTYPE/10
158 CFTYPE = CFTYPE - LDV * 10
159 DIAM(IDX) = AMINI(DIAM(IDX+1),DIAM(IDX-1))
160 GO TO 310
161 C
162 C ***** PROCESS A TAP *****
163 C
164 500 WDOTH(IDX) = WDOTH(IGAS)
165 LDV = CFTYPE/10
166 CFTYPE = CFTYPE - LDV * 10
167 FLD = FRCOEF(IDX)*LOD(IDX)
168 TEMP(IDX) = TEMP(IDX-ISIGN)
169 DIAM(IDX) = AMINI(DIAM(IDX+1),DIAM(IDX-1))
170 C
171 C ***** COMPUTE LINE, CONTROL, FITTING OR TAP DELTA PRESSURE *****
172 C
173 510 IX = IDX - ISIGN

```

```

***** .CHPCAL *****
174      GO TO(520,550),GSTATE
175      C
176      C      ***** DELTA PRESSURE WHEN GASEOUS
177      C
178      C      CALC. RHO OF GAS
179      520 CALL GSDNST (IGAS,TEMP(IX),PRES(IX),RHO)
180      DELP = C1*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IDX)**4)
181      C
182      C      ***** IF PCT. OF PRESSURE CHANGE EXCEEDS ONE PCT. - RECOMPUTE
183      C      ***** DELTA-P, IF NOT, COMPUTE THE NEW PRESSURE
184      C
185      IF(DELP/(PRES(IX) + DELP) - 0.01)560,560,530
186      C
187      C      CALC. RHO OF GAS
188      530 CALL GSDNST (IGAS,TEMP(IX),PRES(IX)+DELP/2.0,RHO)
189      DELP = C1*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IDX)**4)
190      C
191      C      ***** AGAIN CHECK PCT. OF PRESSURE CHANGE. IF PCT. EXCEEDS
192      C      * 2.8 PCT. COMPUTE THE DELTA-P BY USE OF THE COMPRESSIBLE
193      C      ***** FLOW EQUATIONS. (REF.-RPL-TDR-64-25,VOL.1,REV.D)
194      C
195      IF(DELP/(PRES(IX) + DELP) - 0.028)560,560,540
196      C
197      540 A = PI*DIAM(IDX)**2/576.0
198      CALL COMFLO(IDX,PRES(IX),TEMP(IX),FLD,A,WDOTN(IX)/CNOPER,IGAS,
199      1 DELP)
200      PRES(IDX) = PRES(IX) + ISIGN * DELP
201      GO TO 561
202      C
203      C      ***** DELTA PRESSURE WHEN LIQUID
204      C
205      550 CALL RHO LIQ(TEMP(IX),IGAS,RHO)
206      DELP = C1*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IDX)**4)
207      C
208      C      ***** COMPUTE NEW PRESSURE
209      C
210      560 PRES(IDX) = PRES(IX) + ISIGN*DELP
211      C
212      C      ***** COMPUTE THE GAS MACH NUMBER
213      C
214      IF(GSTATE,EQ.2) GO TO 561
215      C
216      C      CALC. RHO OF GAS
217      CALL GSDNST (IGAS,TEMP(IX),PRES(IDX),RHO)
218      C
219      IF(GSTATE,EQ.1) CALL VGV5(IDX,RHO,IGAS)
220      C
221      561 CONTINUE
222      C
223      C      ***** COMPUTE LINE WEIGHT
224      C
225      IF(CFUNCT,EQ.3) GO TO 562
226      IF(CFUNCT,EQ.5) GO TO 562
227      IF(CFUNCT,EQ.6) GO TO 562
228      IF(CFUNCT,EQ.7) GO TO 562
229      IF(CFUNCT,EQ.8) GO TO 562
230      GO TO 570
231      C

```



```

***** CHPCAL *****
232 562 CALL LWEGHT(IDX,LDV)
233 C
234 GO TO 999
235 C
236 ***** COMPUTE CONTROL, FITTING OR TAP WEIGHT
237 C
238 570 WEIGHT(IDX) = CFTW (DIAM(IDX),PRES(IDX),IDV)
239 GO TO 999
240 C
241 ***** PROCESS AN ACCUMULATOR *****
242 C
243 600 PRES(IDX) = APRES(IGAS)
244 TEMP(IDX) = TEMP(IDX - ISIGN)
245 WDOTN(IDX) = WDOTN(IDX - ISIGN)
246 INDXAC(IGAS) = IDX
247 GO TO 999
248 C
249 ***** PROCESS A TANK OR SUPPLY *****
250 C
251 700 CFT = 1
252 INDXTK(IGAS) = IDX
253 C
254 IF(SIPRES(IGAS,CFT)) 720,710,720
255 C
256 ***** IF NO TANK INPUT PRESSURE IS INPUT USE THE VALUE CALC.
257 C
258 710 SIPRES(IGAS,CFT) = PRES(IDX-ISIGN)
259 GO TO 740
260 C
261 ***** CHECK THE TANK INPUT PRESSURE AGAINST THE REQUIRED
262 * CALCULATED PRESSURE. IF THE TANK INPUT PRESSURE IS LESS
263 * THAN THE CALC. PRESSURE WRITE A DIAGNOSTIC MESSAGE AND
264 * SET THE TANK INPUT PRESSURE = THE REQUIRED PRESSURE.
265 ***** IF NOT CONTINUE WITH CALCULATIONS.
266 C
267 720 IF(SIPRES(IGAS,CFT) - PRES(IDX-ISIGN)) 730,740,740
268 C
269 730 WRITE (6,6000) SIPRES(IGAS,CFT), PRES(IDX-ISIGN)
270 GO TO 710
271 C
272 740 PRES(IDX) = SIPRES(IGAS,CFT)
273 C
274 ***** DO THE SAME CHECKS FOR THE INPUT TANK TEMPRATURE.
275 C
276 IF(SITEMP(IGAS,CFT)) 760,750,760
277 750 SITEMP(IGAS,CFT) = TEMP(IDX-ISIGN)
278 C
279 760 TEMP(IDX) = SITEMP(IGAS,CFT)
280 C
281 WDOTN(IDX) = WDOTN(IDX-ISIGN)
282 GO TO 999
283 C
284 ***** PROCESS A PUMP *****
285 C
286 ***** CHECK ISIGN TO SEE IF THE CONDITIONS ON BOTH SIDES OF
287 ***** THE PUMP HAVE BEEN CALCULATED.
288 C
289 800 IF(ISIGN.LT.0) GO TO 825

```

```

***** CHPCAL *****
290      ISIGN = -1
291      C
292      C      ***** SEARCH FORWARD IN LINE FOR A SOURCE TANK
293      C
294      DO 810 I2=IDX,ICNF
295      CALL GETCON(I2)
296      IF(CFUNCT.EQ.I2) GO TO 820
297      810 CONTINUE
298      C
299      WRITE(6,6010) IDX
300      CALL EXIT
301      C
302      C      ***** WHEN A SOURCE TANK IS FOUND, SETUP PRESSURE, TEMPERATURE,
303      C      ***** FLOW RATE AND FLAG TO CONTINUE THE CALCULATIONS.
304      C
305      820 WDOTN(I2) = WDOTN(IDX-1)
306      IOT = I2 - 1
307      IDX = I2
308      C
309      PRES(IDX) = SIPRES(IGAS,I)
310      TEMP(IDX) = SITEMP(IGAS,I)
311      GO TO 1000
312      C
313      C      ***** COMPUTE THE WEIGHT OF THE PUMP, TURBINE, PROPELLANT AND
314      C      ***** MOTOR DEPENDING ON THE TYPE OF PUMP.
315      C
316      C      ***** CHECK CFTYPE FOR HIGH OR LOW PRESSURE PUMP
317      C
318      825 ISIGN = 1
319      PRES(IDX) = PRES(IDX+1)
320      TEMP(IDX) = TEMP(IDX+1)
321      WDOTN(IDX) = WDOTN(IDX+1)
322      C
323      JOPTN = CFTYPE/10
324      CFTYPE = CFTYPE - JOPTN * 10
325      C
326      IF(CFTYPE.EQ.2) GO TO 840
327      C
328      C      ***** PROCESS THE HIGH PRESSURE PUMP
329      C
330      C
331      C      ***** COMPUTE THE PUMP OR TURBOPUMP WEIGHT
332      C
333      JJOPTN(IGAS) = JOPTN
334      PTEMP(IGAS) = TEMP(IDX)
335      PPRES(IGAS) = PRES(IDX)
336      PPDCH(IGAS) = PRES(IDX-1)
337      PDELP = ABS(PRES(IDX-1) - PRES(IDX))
338      PPDEL(IGAS) = PDELP
339      PWDOT = WDOTN(IDX)/CHOPER
340      PPWDOT(IGAS) = PWDOT
341      CALL RHOLIO(TEMP(IDX),IGAS,RHO)
342      PPRHO(IGAS) = RHO
343      C      CALCULATE PUMP PARAMETERS
344      C
345      CALL PARPHP(IGAS,JOPTN,PDELP,PPWDOT,PNPSH(IGAS),RHO,PMEF,V,
346      I E, WT, PNSG, NSTG, NPSPR)
347      PMPEFF(IGAS) = PMEF

```

```

***** CHPCAL *****
348      PMPVOL(IGAS) = V
349      PMPW(IGAS) = E
350      PSPD(IGAS) = PMSG
351      PSTAGE(IGAS) = NSTG
352      PNPSPR(IGAS) = NPSPR
353      PWEIGHT(IGAS) = WT * (CNOPEP + CNSTBY)
354      C
355      C      ***** CHECK PTYPE FOR PUMP OR TURBOPUMP *****
356      C
357      IF(PTYPE.EQ.1) GO TO 830
358      C
359      C      ***** COMPUTE THE TURBINE WEIGHT
360      C
361      CALL TURBN(IGAS,TRBWGT)
362      TWEGHT(IGAS) = TRBWGT * (CNOPEP + CNSTRY)
363      C
364      C      ***** COMPUTE THE FLOWRATE OF THE GAS GENERATOR AND ITS WEIGHT
365      C
366      TMEAN1 = (TITEMP(1) - TOTEMP(1))/2.0
367      TMEAN2 = (TITEMP(2) - TOTEMP(2))/2.0
368      CALL CSUBP1(TMEAN1,THRATO(1),CPEP1)
369      CALL CSUBP1(TMEAN2,THRATO(2),CPEP2)
370      DLHTP1 = CPEP1 * (TITEMP(1) - TOTEMP(1))
371      DLHTP2 = CPEP2 * (TITEMP(2) - TOTEMP(2))
372      CALL RHOLIO(SITEMP(1,1),1,RHOLQ1)
373      CALL RHOLIO(SITEMP(2,1),2,RHOLQ2)
374      BRAC1 = (0.185 * PPDEL(1))/(RHOLQ1 * PHPEFF(1) * TEFF(1) * DLHTP1)
375      C1 = THRATO(1)/(1.0 + THRATO(1))
376      C2 = 1.0/(1.0 + THRATO(1))
377      IF(PPDEL(2).EQ.0.0) PPDEL(2) = EPDELP(2)
378      IF(PHPEFF(2).EQ.0.0) PHPEFF(2) = PEFF(2)
379      BRAC2 = (0.185 * PPDEL(2))/(RHOLQ2 * PHPEFF(2) * TEFF(2) * DLHTP2)
380      C3 = THRATO(2)/(1.0 + THRATO(2))
381      C4 = 1.0/(1.0 + THRATO(2))
382      C5 = BRAC1 * C1
383      C6 = BRAC1 * C2
384      C7 = BRAC2 * C3
385      C8 = BRAC2 * C4
386      D1 = (C5 + C8 + C6*C7 - C5*C8 - 1.0)
387      WFL01 = (WDOTI(1) * (C5*C8 - C5) + WDOTI(2) * (-C5*C7))/D1 * CNOPEP
388      WFL02 = (WDOTI(1) * (-C6*C7) + WDOTI(2) * (C5*C7 - C7))/D1 * CNOPEP
389      WFL03 = (WDOTI(1) * (C6*C8 - C6) + WDOTI(2) * (-C6*C7))/D1 * CNOPEP
390      WFL04 = (WDOTI(1) * (-C6*C8) + WDOTI(2) * (C5*C8 - C8))/D1 * CNOPEP
391      WGGGFR(1) = (WFL01 + WFL03) * CNOPEP
392      WGGGFR(2) = (WFL02 + WFL04) * CNOPEP
393      C
394      KK = 0
395      WDOTG1 = 0.0
396      WDOTG2 = 0.0
397      C
398      DO 835 I2 = 1,NDCYCL*2
399      KK = KK + 1
400      WFL05 = (WDOTJ(KK,1) * (C5*C8 - C5) + WDOTJ(KK,2) * (-C5*C7))/D1
401      WFL06 = (WDOTJ(KK,1) * (-C6*C7) + WDOTJ(KK,2) * (C5*C7 - C7))/D1
402      WFL07 = (WDOTJ(KK,1) * (C6*C8 - C6) + WDOTJ(KK,2) * (-C6*C7))/D1
403      WFL08 = (WDOTJ(KK,1) * (-C6*C8) + WDOTJ(KK,2) * (C5*C8 - C8))/D1
404      WFL057 = (WFL05 + WFL07)/CNOPEP
405      WFL068 = (WFL06 + WFL08)/CNOPEP

```

```

***** CHPCAL *****
406      WDOTG1 = WDOTG1 + WFLO57 * DCYCLE(I2)
407      WDOTG2 = WDOTG2 + WFLO68 * DCYCLE(I2)
408      835 CONTINUE
409      C
410      GWEGHT(1) = WDOTG1 * CNOPER
411      C
412      GWEGHT(2) = WDOTG2 * CNOPER
413      C
414      ***** COMPUTE SYSTEM WEIGHT
415      C
416      ATERM = 13.824204 - (0.01117823*TGGPC( IGAS )) + (1.8632927E-5 *
417      1(TGGPC( IGAS )**2)) - (1.108423E-8 * (TGGPC( IGAS )**3))
418      C
419      BTERM = 7.9470262 - (.035636198*TGGPC( IGAS )) + (6.4684644E-5 *
420      1(TGGPC(IGAS )**2)) - (3.7946E-8 * (TGGPC( IGAS )**3))
421      C
422      WTGGA = ATERM + BTERM * WDGGFR(IGAS)/CNOPER
423      WGTGGA(IGAS) = WTGGA * (CNOPER + CNSTBY)
424      HEIGHT(IDX) = PWEGHT(IGAS) + TWEGHT(IGAS) + WGTGGA(IGAS)
425      C
426      GO TO 999
427      C
428      C
429      830 CONTINUE
430      HEIGHT(IDX) = PWEGHT(IGAS)
431      C
432      GO TO 999
433      C
434      C
435      C
436      C
437      840 CONTINUE
438      IF((SYSNUM.EQ.2.OR.SYSNUM.EQ.4) .AND. (SCRIT.EQ.2)) GO TO 841
439      CALL FINTAB (HTBID(15)+IGAS)
440      XTAB(1) = TPEFF(IGAS)*100.0
441      XTAB(2) = TPMPSH(IGAS)
442      IF(TPDELP(IGAS))860,850,860
443      850 TPDELP(IGAS) = PRES(IDX) - PRES(IDX-1)
444      860 IF(TPWDOT(IGAS))880,870,880
445      870 TPWDOT(IGAS) = WDOTN(IDX)/CNOPER
446      880 XTAB(3) = TPDELP(IGAS)
447      XTAB(4) = TPWDOT(IGAS)
448      TPWGHT(IGAS) = MIPE(4,XTAB)*(CNOPER + CNSTBY)
449      C
450      C
451      C
452      ***** ELECTRIC MOTOR FOR TRANSFER PUMP
453      C
454      CALL FINTAB (HTBID(16))
455      CALL RHOLIO(TEMP(IDX),IGAS,RHO)
456      HP = 144.0*WDOTN(IDX)*(PRES(IDX) - PRES(IDX-1))/(350.0*TPEFF*RHO*
457      NEFF)
458      XTAB(1) = HP
459      XTAB(2) = HSS
460      EMHGT = MIPE(2,XTAB) * (CNOPER + CNSTBY)
461      WEIGHT(IDX) = EMHGT + TPWGHT(IGAS)
462      C
463      KK = 0
464      BWEGHT(IGAS) = 0.0

```

```

***** CHPCAL *****
464      DO 890 I2 = 1, NDCYCL*2
465      KK = KK + 1
466      HP = 144.0*WDOTJ(KK, IGAS)*(PRES(IDX) - PRES(IDX - 1))/(550.0*PEFF*
467      1 RHO*MEFF)
468      PB = HP*746.0*DCYCLE(I2)/3600.0
469      BWEGHT(IGAS) = BWEGHT(IGAS) + PB/PDNSTY
470      890 CONTINUE
471      GO TO 999
472  C
473      841 CONTINUE
474      WEIGHT(IDX) = WCIRCP(IGAS)
475      GO TO 999
476  C
477  C      ***** PROCESS A HEAT EXCHANGER *****
478  C
479      900 IF (ISIGN.GT.0) GO TO 910
480  C
481      WRITE (IOT,6005) ISIGN
482  C
483      910 CONTINUE
484      JX = JX + 1
485      JHX = JX
486      WDOTN(IDX) = WDOTN(IDX-ISIGN)
487      WDOTCF(JX, IGAS) = WDOTN(IDX)
488      UCODE(JX, IGAS) = CODE(IDX)
489  C
490      HXDLP = HEXCIP(JX, IGAS) - HEXCOP(JX, IGAS)
491  C
492      IF (HXDLP.GT.0.0) GO TO 911
493      UCODE(JX, IGAS) = 6HNONE
494      PRES(IDX) = PRES(IDX-1)
495      TEMP(IDX) = TEMP(IDX-1)
496      GO TO 1000
497  C
498      911 CONTINUE
499  C
500      HEXCOP(JX, IGAS) = PRES(IDX - 1)
501      HEXCIP(JX, IGAS) = HEXCOP(JX, IGAS) + HXDLP
502      IF (SCRIT.EQ. 1) GO TO 913
503      IF (SYSNUM.EQ. 2 .OR. SYSNUM.EQ. 4) GO TO 912
504      913 CONTINUE
505  C
506  C      COMPUTE HEATEX PARAMETERS
507  C
508      CALL HEATEX(IGAS, JHX, WDOTN(IDX), HEXHIT(JX, IGAS), HEXCIT(JX, IGAS),
509      1 HEXHOT(JX, IGAS), HEXCOT(JX, IGAS), HEXHIP(JX, IGAS), HEXCIP(JX, IGAS),
510      2 HEXHOP(JX, IGAS), HEXCOP(JX, IGAS), HXMRAT(JX, IGAS), WDOTH(JX, IGAS),
511      3 WHXTOT(JX, IGAS))
512  C
513  C
514  C      ***** COMPUTE THE GAS GENERATOR ASSEMBLY WEIGHT *****
515  C
516      CALL GASGEN(JX, IGAS)
517  C
518      912 CONTINUE
519  C
520      TEMP(IDX) = HEXCIT(JX, IGAS)
521      DLPRES = HEXCIP(JX, IGAS) - HEXCOP(JX, IGAS)

```

```

***** .CHPCAL *****
522      PRES(IDX) = PRES(IDX-ISIGN) + DLPRES*ISIGN
523      WEIGHT(IDX) = WHXTOT(JX,IGAS)
524      C
525      C      ***** END OF CONFIGURATION PROCESSING LOOP *****
526      C
527      999 CONTINUE
528      IF (.NOT. PAGE(1)) GO TO 1998
529      C
530      C      *** PAGE HEADER HAS BEEN MOVED TO STATEMENT GROUP 100
531      C
532      1998 CONTINUE
533      KFUNCT = FNAME(CFUNCT)
534      WRITE(IOT,6030)KFUNCT,CODE(IDX),CFTYPE,CNOPER,CNSTBY,ISIGN,IDX,
535      1      IGAS,GSTATE,FCOEFF(IDX),LOD(IDX),DIAM(IDX),
536      2      ITTHICK(IDX),PRES(IDX),TEMP(IDX),WDOTN(IDX),
537      3      WEIGHT(IDX),MACH(IDX),MFLG(IDX)
538      IF (PRES(IDX) .GE. 0. .AND. TEMP(IDX) .GE. 0.) GO TO 998
539      WRITE(IOT,6040)
540      CALL EXIT
541      998 IF (CFUNCT.EQ. 13) IDX = 1DT
542      C
543      270 CONTINUE
544      1000 CONTINUE
545      1100 CONTINUE
546      C      USED BY WEIGHT SUMMARY OUTPUT
547      KEND = IDX - 1
548      KOEND = IHSTT - 2
549      C
550      IF((SYSNUM.EQ.2.OR.SYSNUM.EQ.4) .AND. (SCRIT.EQ.2)) GO TO 1150
551      C
552      C      OUTPUT THE HEAT EXCHANGER AND GAS GENERATOR DATA      *****
553      C
554      CALL OTPHEX
555      C
556      C      OUTPUT THE PUMP AND TURBINE DATA      *****
557      C
558      CALL OTPPMP
559      C
560      CALL OTPTRB
561      C
562      1150 CONTINUE
563      C
564      RETURN
565      C
566      C      ***** OUTPUT FORMATS
567      C
568      6000 FORMAT(10 *DIAGNOSTIC* TANK INPUT PRESSURE IS LESS THAN THE REQUIR
569      1ED PRESSURE. TANK PRESSURE SET = REQUIRED PRESSURE,1/15X,TANK INP
570      2UT PRESSURE = 1,F7.2,1, REQUIRED PRESSURE = 1,F7.2)
571      C
572      6005 FORMAT(10 **ERROR** ISIGN =1,13,1THERE IS A CONFIGURATION ERROR1/)
573      C
574      6010 FORMAT(10 *ERROR* A PUMP WAS ENCOUNTERED BUT NO TANK CAN BE FOUND.
575      1 PUMP CONFIGURATION INDEX NUMBER = 1,13)
576      C
577      6020 FORMAT(10 F CODE FT NO NS IS IDX G GS FCOEF L/D
578      1 DIAH ITTHICK PRES TEMP WDOT WEIGHT MACH M
579      2FLAG1/1 1)

```

```

***** CHPCAL *****
580 C
581 6030 FORMAT(2XA3,2XA6,I3,6I4,F9.6,F12.4,2F8.4,2F9.2,2F7.2,F9.3,F10.7,
582 13X,A6)
583 C
584 6040 FORMAT (T44, '*** TERMINATE - NEGATIVE TEMP. OR PRES. ***')
585 C
586 6050 FORMAT(/T38,'*** SUMMARY OF COMPUTED SYSTEM CONFIGURATION PARAMETE
587 IRS ***')
588 C
589 6051 FORMAT(/T32,'*** SUMMARY OF COMPUTED SYSTEM CONFIGURATION PARAMETE
590 IRS - CONTD. ***')
591 C
592 END

```

PROCEDURE DEFINITION PROCESSOR - CNAMES

```
1  CNAMES* PROC
2  C
3      INTEGER FNAME
4  C
5      COMMON /CNAMES/ FNAME(18), L0(9,15), L1(21,2), L2(3,7), L3(4,3),
6      1 L4(4,11), L5(4,5), L6(4,4), L7(4,23), L8(4,3), L9(4,5), L10(4,5),
7      2 L11(4,20), L12(4,28), L13(4,28), JFLUID(2,3), KFLUID(2,2)
8  C
9      C      NAMES ARE GIVEN IN S.R. STODTA
10 C
11 END
```


***** SUBROUTINE COMFLØ

```

1 C * * * * *
2 C * ROUTINE NAME - COMPRESSIBLE FLOW EQUATIONS *
3 C * EVALUATION ROUTINE *
4 C * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
5 C * PROGRAMMER - R. BOLLINGER 1943 102 26933 *
6 C * DATE CODED - 3/10/70 *
7 C * * * * *
8 C
9 SUBROUTINE COMFLO(IDX,P,T,FLD,A,WDOT,N,DELP)
10 C
11 C ***** EXPLANATION OF THE CALLING SEQUENCE
12 C *
13 C * P = PRESSURE
14 C *
15 C * T = TEMPRATURE
16 C *
17 C * FLD = RESISTANCE COEFFICIENT
18 C *
19 C * A = AREA
20 C *
21 C * WDOT = FLOW RATE
22 C *
23 C * N = GAS TYPE NUMBER
24 C * N = 1 FOR OXYGEN
25 C * N = 2 FOR HYDROGEN
26 C * 17 FOR HE
27 C *
28 C ***** DELP ..COMPUTED PRESSURE DROP
29 C
30 LOGICAL DIAG
31 C
32 REAL M1, M2, MIMIN, MIMAX
33 C
34 INCLUDE CCNFIG
35 INCLUDE CIOUNT
36 INCLUDE CONST
37 C
38 DATA JBLNK,JAST1,JAST6/' ',,' ',,'*****'/
39 C
40 HFLG(IDX) = JBLNK
41 C
42 C ***** COMPUTE GAMMA = SPECIFIC HEAT AT CONSTANT PRESSURE/
43 C * SPECIFIC HEAT AT CONSTANT VOLUME
44 C * *****
45 C
46 IF (DIAG(0,6HCOMFLO)) WRITE (10T,6010) N,P,T,FLD,A,WDOT,DELP
47 IF(N.EQ.2) GO TO 60
48 CALL CSUBP(T,P,N,CP)
49 CPI = CP
50 CVI = CSUBV(T,P,N)
51 GMA = CPI/CVI
52 GO TO 65
53 C
54 60 GMA = HPTGAM(P,T)
55 C
56 65 CONTINUE
57 C

```

```

***** COMFLO *****
58 C ***** COMPUTE MACH NUMBER AT PRIOR CONDITIONS
59 C
60 M2 = WDOT * SQRT(FINDR(N)*7/(GMA*GRAVTV))/(P*A*144.0)
61 C
62 C ***** COMPUTE RESISTANCE COEFFICIENT AT PRIOR CONDITIONS
63 C
64 FLDMAX = FLODEQ(M2,GMA)
65 DF = FLDMAX + FLD
66 C
67 C ***** ITERATE TO FIND MACH NUMBER AT CURENT CONDITIONS
68 C
69 MIMIN = 1.E-10
70 MIMAX = 1.0
71 C
72 DO 40 I1 = 1,20
73 M1 = (MIMAX + MIMIN)/2.0
74 FAC = FLODEQ(M1,GMA) - DF
75 IF (ABS(FAC) - 0.00005) 50,50,10
76 10 IF (FAC) 20,50,30
77 20 MIMAX = M1
78 GO TO 40
79 30 MIMIN = M1
80 40 CONTINUE
81 C
82 WRITE (6,6000) M1
83 C
84 C ***** COMPUTE THE PRESSURE DROP
85 C
86 50 A1 = M1**2
87 A2 = M2**2
88 P20P1 = (M1/M2) * (2.0 + (GMA - 1.0) * A1)/(2.0 + (GMA - 1.0) * A2)
89 DELP = P * (1.0/P20P1 - 1.0)
90 C
91 C ***** SET FINAL MACH NUMBER FROM ITERATION
92 C
93 MACH(IDX) = M1
94 C
95 C ***** CHECK MACH NUMBER, FLAG MACH NUMBER GREATER THAN 0.3
96 C * WITH ONE ASTERISK, FLAG MACH NUMBER GREATER THAN
97 C ***** 1.0 WITH SIX ASTERISKS.
98 C
99 IF (MACH(IDX) - 0.3) 80,80,81
100 81 IF (MACH(IDX) - 1.0) 82,83,83
101 82 MFLG(IDX) = JAST1
102 GO TO 80
103 83 MFLG(IDX) = JAST6
104 80 CONTINUE
105 IF (DIAG(1,6HCOMFLO)) WRITE (10,6010) I1,GMA,CP,FLDMAX,M1,M2,DELP
106 RETURN
107 C
108 C ***** DIAGNOSTIC MESSAGE
109 C
110 6000 FORMAT('0 *DIAGNOSTIC* MACH NUMBER COMPUTATION FAILED TO CONVERGE
111 WHILE COMPUTING COMPRESSABLE FLOW EQUATIONS. MACH SET = 1.E12.7)
112 C
113 6010 FORMAT('1+'14X,15,10X,6E15.6)
114 END

```

***** SUBROUTINE COMPIL

```

1      C      * * * * *
2      C      * ROUTINE NAME = DATA INPUT, VERIFY AND ECHO *
3      C      * ROUTINE LANG = FORTRAN V UNIVAC 1108 EXEC 2 *
4      C      * PROGRAMMER  = R. BOLLINGER 1943 102 26933 *
5      C      * DATE CODED  = APRIL FOOLS DAY 1970      *
6      C      * REVISED   = JANUARY 11, 1972          *
7      C      * REVISED   = JULY 1972                 *
8      C      * PROGRAMMER  = J. MCKAY D1943 201 45178 *
9      C      * * * * *
10     C
11     C      SUBROUTINE COMPIL
12     C
13     C      LOGICAL JP,PAGE
14     C
15     C      INCLUDE CACUM
16     C      INCLUDE CAPI
17     C      INCLUDE CCNFIG
18     C      INCLUDE CCNTRL
19     C      INCLUDE CDCYCL
20     C      INCLUDE CECLSS
21     C      INCLUDE CENG
22     C      INCLUDE CFUEL
23     C      INCLUDE CHEX
24     C      INCLUDE CHSARC
25     C      INCLUDE CIOUNT
26     C      INCLUDE CMOTOR
27     C      INCLUDE CNAMES
28     C      INCLUDE CPAGE
29     C      INCLUDE CRUMP
30     C      INCLUDE CTANK
31     C      INCLUDE CTURBN
32     C      INCLUDE TANKWT
33     C
34     C      5010 FORMAT(A6,I4,3I5,3F5.0,I5,2F5.0,5X,A6)
35     C      5020 FORMAT(15,6F10.0)
36     C      5030 FORMAT(3I5/7F10.0/F10.0)
37     C      5039 FORMAT(15)
38     C      5040 FORMAT(11F6.0,6X,A6)
39     C      5050 FORMAT(15,4F10.0/I5,4F10.0/4F10.0/4F10.0/5F10.0/5F10.0)
40     C      5060 FORMAT(5I5/8F10.0/8F10.0/F10.0)
41     C      5062 FORMAT(2I5,7F10.0)
42     C      5070 FORMAT(F10.0)
43     C      5080 FORMAT(3F10.0,I5,3F10.0,F7.0)
44     C      5090 FORMAT(15,5X,4F10.0)
45     C      5100 FORMAT(15,5X3F10.0)
46     C      5110 FORMAT(8F10.0)
47     C      5120 FORMAT(15,5X,5F10.0)
48     C      5130 FORMAT(7F10.0)
49     C      5140 FORMAT(10F7.0)
50     C      5141 FORMAT(2I5,6F10.0)
51     C      5150 FORMAT(4I5,5F10.0/(7F10.0))
52     C      5151 FORMAT(7F10.0/5F10.0)
53     C
54     C      6000 FORMAT(10I,38X9A6//21A6/21A6/' ')
55     C      6010 FORMAT(3XA6, 2XA6,1XI4,2XI5,2XI5,2XI5,2X,E15,8,6XF7.2,6XF6.2,5XI5,
56     C      1 9XF6.2,7XF5.1)
57     C      6020 FORMAT('0 *DIAGNOSTIC* THE ABOVE FUNCTION CODE IS ILLEGAL')

```

```

*****
***** COMPIL *****
58 6030 FORMAT(10,38X9A6//)
59 6031 FORMAT(10,T55,'SUBCRITICAL APU DATA'//)
60 6032 FORMAT(10,T53,'SUPERCRITICAL APU DATA'//)
61 6040 FORMAT(54X15,2X3A6/(46XE13.8,2X3A6))
62 6050 FORMAT(10,38X9A6//)
63 6051 FORMAT(T18,1- 1 -1,T36,1- 2 -1,T54,1- 3 -1,T72,1- 4 -1,T90,1- 5 -1
64 1 ,T103,'HEAT EXCHANGER NUMBER'/T13,'OXYGEN HYDROGEN OXYGEN HYDR
65 2OGEN OXYGEN HYDROGEN OXYGEN HYDROGEN OXYGEN HYDROGEN'//)
66 6052 FORMAT(////T18,1- 6 -1,T36,1- 7 -1,T54,1- 8 -1,T72,1- 9 -1,T90,
67 1 -10 -1,T103,'HEAT EXCHANGER NUMBER'/T13,'OXYGEN HYDROGEN OXYG
68 2EN HYDROGEN OXYGEN HYDROGEN OXYGEN HYDROGEN OXYGEN HYDROGEN
69 3'//)
70 6060 FORMAT (41X15,10X15,7X4A6)
71 6062 FORMAT ('0'//38X1*9A6/ 39X1TANK'8X1FLUID'9X1. . . D I M E N S I O
72 1 N S . . . / 39X1SHAPE'7X1TYPE'22X1(FEET)'//)
73 6064 FORMAT (30X 21I2,6X 3F12,4)
74 6069 FORMAT(T48,'NUMBER OF HEAT EXCHANGERS INPUT ='1,15//)
75 6070 FORMAT(9X10F9,1,3X4A6)
76 6080 FORMAT(42X14,10X15,7X4A6/4(36X2E15.8,2X4A6//),10,38X9A6//4(36X2E1
77 15.8,2X4A6//),10,38X9A6// (36X2E15.8,2X4A6))
78 6090 FORMAT (36X2E15.8,2X4A6)
79 6091 FORMAT(10,47X,15,15X,'TANK HEIGHT-CONFIGURATION OPTION CONSIDERED
80 1 '/48X,15,15X,'NUMBER OF TANK SHAPES IN CONFIGURATION')
81 6100 FORMAT(10,38X9A6//)
82 6101 FORMAT(10,T10,'OPER. TIME',T24,'NON-OPERATING',T40,'MIB-DEGRAD.',
83 1 T54,'UNITS OPER.',T67,'HORSEPOWER',T81,'AMB.PRESSURE',T98,
84 2 'POWER-KW',T110,'REPPES, TIME'//)
85 6110 FORMAT(T7,3E15.8,1R,5X,4E15.8)
86 6120 FORMAT ('0 * ERROR * DUTY CYCLE INPUT TOO LONG'//)
87 6127 FORMAT(//T18,1- 6 -1,T36,1- 7 -1,T54,1- 8 -1,T72,1- 9 -1,T90,
88 1 -10 -1,T103,'HEAT SOURCE NUMBER'/T13,
89 2 'OXYGEN HYDROGEN OXYGEN HYDR
90 3OGEN OXYGEN HYDROGEN OXYGEN HYDROGEN OXYGEN HYDROGEN'//)
91 6128 FORMAT(T49,'NUMBER OF HEAT SOURCES INPUT ='1,15//)
92 6129 FORMAT(T18,1- 1 -1,T36,1- 2 -1,T54,1- 3 -1,T72,1- 4 -1,T90,1- 5 -1
93 1 ,T103,'HEAT SOURCE NUMBER'/T13,'OXYGEN HYDROGEN OXYGEN HYDROGEN
94 2 OXYGEN HYDROGEN OXYGEN HYDROGEN OXYGEN HYDROGEN'//)
95 6130 FORMAT(9X10I9,3X4A6/(9X10F9,1,3X4A6))
96 6140 FORMAT(T52,1I5,T70,'MOTOR TYPE',T52,E15.8,T70,'MOTOR EFFICIENCY',
97 1 /T52,E15.8,T70,'MOTOR SPEED',T52,E15.8,T70,'BATTERY POWER DENSITY
98 2')
99 6150 FORMAT(54X15,2X4A6/(46XE13.8,2X4A6))
100 6160 FORMAT(54X15,2X4A6/54X15,2X4A6/54X15,2X4A6/54X15,2X4A6,
101 1 /(46XE13.8,2X4A6))
102 C
103 C *****
104 C
105 C ***** CHECK 'IBLK' BEFOR DATA SETUP - INCLUDE ALL DATA BLOCKS *****
106 C ***** REQUIRED FOR THE SYSTEM BEING STUDIED. *****
107 C
108 C *****
109 C
110 C NIENTH = 0.0
111 C
112 C
113 C ***** INPUT THE CONFIGURATION TABLE
114 C
115 C IF(PAGE(0)) WRITE(6,6000) (LO(1,1),I=1,9),((LI(1,J),I=1,21),J=1,2)

```

```

***** COMPIL *****
116      JP = PAGE(5)
117      C
118      IERR = 0
119      DO 30 I1=1,ICNF
120      READ (IIN,5010) CFUNCT, CFTYPE, CNOPER, CNSTBY, CHTYPE, FRCOEF(I1)
121      1      , LOD(I1), DIAM(I1), CITYPE, ITHICK(I1), NBAR(I1),
122      2      CODE(I1)
123      IF(PAGE(1)) WRITE(6,6000) (LO(I1,I),I=1,9),((LI(I1,J),J=1,2),J=1,2)
124      WRITE (IOT,6010) CFUNCT, CODE(I1), CFTYPE, CNOPER, CNSTBY, CHTYPE,
125      1      FRCOEF(I1), LOD(I1), DIAM(I1), CITYPE,
126      2      ITHICK(I1), NBAR(I1)
127      C
128      C      ***** SEARCH FOR THE CONFIGURATION NAME.
129      C
130      DO 10 I2 = 1,18
131      IF(CFUNCT.EQ.FNAME(I2)) GO TO 20
132      10 CONTINUE
133      C
134      WRITE(6,6020)
135      IERR = IERR + 1
136      GO TO 30
137      C
138      20 CFUNCT = 12
139      CALL STOCOH(I1)
140      IF(CFUNCT.EQ.18) GO TO 35
141      30 CONTINUE
142      C
143      C      ***** INPUT THE DUTY CYCLE DATA
144      C
145      35 CONTINUE
146      IF (IERR .GT. 0) CALL EXIT
147      IF(PAGE(0)) WRITE(6,6100) (LO(I,6),I=1,9)
148      WRITE (IOT,6101)
149      DCYCLT = 0.0
150      NDCYCL = 0
151      II = 0
152      DO 100 I1=1,ICDL2,2
153      NDCYCL = NDCYCL + 2
154      II = II + 1
155      READ (IIN,5080) DCYCLE(I1), DCYCLE(I1+1), PSI(I1), NEOP(I1),
156      1      HP(I1), PAMB(I1), PKW(I1), RPRTIM(I1)
157      IF(PAGE(1)) WRITE(6,6100) (LO(I,6),I=1,9)
158      WRITE (IOT,6110) DCYCLE(I1), DCYCLE(I1+1), PSI(I1), NEOP(I1),
159      1      HP(I1), PAMB(I1), PKW(I1), RPRTIM(I1)
160      DCYCLT = DCYCLT + DCYCLE(I1)
161      KCYCLE = II - 1
162      IF(DCYCLE(II+1)) 90,,
163      IF(DCYCLE(II)),100,100
164      NDCYCL = NDCYCL - 1
165      90 NDCYCL = NDCYCL - 1
166      GO TO 110
167      100 CONTINUE
168      C
169      WRITE(6,6120)
170      CALL EXIT
171      C
172      C      ***** INPUT THE CONSUMER DATA
173      C

```

***** COMPIL *****

```

174      110 CONTINUE
175      GO TO (111,112,120,130,111), SYSNUM
176      111 CONTINUE
177      C
178      C      ***** READ IN THE ENGINE CONSUMER DATA
179      C
180      IF (PAGE(0)) WRITE(6,6030) (LO(I,2),I=1,9)
181      READ(5,5020) HENG,GITEMP,GIPRES,THRUST,PSUBC,EXPRAT,MIXRAT
182      WRITE(6,6040) HENG,(L2(I,1),I=1,3),
183      1      GITEMP,(L2(I,2),I=1,3),
184      2      GIPRES,(L2(I,3),I=1,3),
185      3      THRUST,(L2(I,4),I=1,3),
186      4      PSUBC,(L2(I,5),I=1,3),
187      5      EXPRAT,(L2(I,6),I=1,3),
188      6      MIXRAT,(L2(I,7),I=1,3)
189      C
190      GO TO 113
191      C
192      112 CONTINUE
193      C
194      C      *** READ IN THE APU CONSUMER DATA
195      C
196      IF (PAGE(0)) WRITE (IOT,6030) (LO(I,13),I=1,9)
197      C
198      READ (IIN,5120) NAPU, HPR, FMR, PGG, TIT, TD
199      GO TO (114,115),SCRIT
200      114 CONTINUE
201      WRITE (IOT,6031)
202      READ (IIN,5130) MRGGCH, MRGGCO, TDGGH, TDGGO, TVH, TVO, TENV
203      C
204      WRITE (IOT,6150) NAPU, (L11(I,1),I=1,4), HPR, (L11(I,2),I=1,4),
205      1      FMR, (L11(I,3),I=1,4), PGG, (L11(I,4),I=1,4),
206      2      TIT, (L11(I,5),I=1,4), TD, (L11(I,6),I=1,4),
207      3      MRGGCH, (L11(I,7),I=1,4), MRGGCO, (L11(I,8),I=1,4),
208      4      TDGGH, (L11(I,9),I=1,4), TDGGO, (L11(I,10),I=1,4),
209      5      TVH, (L11(I,11),I=1,4), TVO, (L11(I,12),I=1,4),
210      6      TENV, (L11(I,13),I=1,4)
211      C
212      GO TO 113
213      C
214      115 CONTINUE
215      WRITE (IOT,6032)
216      READ (IIN,5130) FMRG, PFH, PFO, TFH, TFO, TG, DELPCP, TENV
217      C
218      WRITE (IOT,6150) NAPU, (L11(I,1),I=1,4), HPR, (L11(I,2),I=1,4),
219      1      FMR, (L11(I,3),I=1,4), PGG, (L11(I,4),I=1,4),
220      2      TIT, (L11(I,5),I=1,4), TD, (L11(I,6),I=1,4),
221      3      FMRG, (L11(I,14),I=1,4), PFH, (L11(I,15),I=1,4),
222      4      PFO, (L11(I,16),I=1,4), TFH, (L11(I,17),I=1,4),
223      5      TFO, (L11(I,18),I=1,4), TG, (L11(I,19),I=1,4),
224      6      DELPCP, (L11(I,20),I=1,4), TENV, (L11(I,13),I=1,4)
225      C
226      GO TO 113
227      C
228      120 CONTINUE
229      C
230      C      *** READ IN LIFE SUPPORT SYSTEM CONSUMER DATA
231      C

```

***** COMPIL *****

```

232      IF(PAGE(0)) WRITE (IOT,6050) (LO(I,15),I=1,9)
233      C
234      READ (IIN,5150) MDAYS,NCREW,NRPRES,NDARES,O2FNOM,GLKRAT,TLNOM(1),
235      1 TLNOM(2),RHOBEG(1),RHOBEG(2),TKFTEM(1),TKFTEM(2),TKFPRS(1),
236      2 TKFPRS(2),TENVR,CABVOL
237      C
238      READ (IIN,5151) LINDIA(1), LINDIA(2), HTRFLX(1), HTRFLX(2),
239      1 PLSNOM(1), PLSNOM(2), HTRDIA(1), HTRDIA(2),
240      2 HTRLNG(1), HTRLNG(2), PSET1, PSET2
241      C
242      WRITE (IOT,6160) MDAYS,(L13(I, 1),I=1,4),
243      1 NCPEW   ,(L13(I, 2),I=1,4),NRPRES   ,(L13(I, 3),I=1,4),
244      2 NDARES  ,(L13(I, 4),I=1,4),O2FNOM   ,(L13(I, 5),I=1,4),
245      3 GLKRAT  ,(L13(I, 6),I=1,4),TLNOM(1),(L13(I, 7),I=1,4),
246      4 TLNOM(2),(L13(I, 8),I=1,4),RHOBEG(1),(L13(I, 9),I=1,4),
247      5 RHOBEG(2),(L13(I,10),I=1,4),TKFTEM(1),(L13(I,11),I=1,4),
248      6 TKFTEM(2),(L13(I,12),I=1,4),TKFPRS(1),(L13(I,13),I=1,4),
249      7 TKFPRS(2),(L13(I,14),I=1,4),TENVR   ,(L13(I,15),I=1,4),
250      8 CABVOL  ,(L13(I,16),I=1,4),LINDIA(1),(L13(I,17),I=1,4),
251      9 LINDIA(2),(L13(I,18),I=1,4),HTRFLX(1),(L13(I,19),I=1,4),
252      T HTRFLX(2),(L13(I,20),I=1,4),PLSNOM(1),(L13(I,21),I=1,4),
253      A PLSNOM(2),(L13(I,22),I=1,4),HTRDIA(1),(L13(I,23),I=1,4),
254      B HTRDIA(2),(L13(I,24),I=1,4),HTRLNG(1),(L13(I,25),I=1,4),
255      C HTRLNG(2),(L13(I,26),I=1,4),PSET1   ,(L13(I,27),I=1,4),
256      D PSET2   ,(L13(I,28),I=1,4)
257      C
258      GO TO 113
259      C
260      130 CONTINUE
261      C
262      *** READ IN FUEL CELL CONSUMER DATA
263      C
264      IF(PAGE(0)) WRITE(IOT,6050) (LO(I,14),I=1,9)
265      C
266      READ (IIN,5140) MRFC, SRCFC, ODTFC, SPWTFC, TFCNOM(1),TFCNOM(2),
267      1 TF2IIN, TF2IOU, TFOFC, TFHFC
268      READ (IIN,5140) PFOFC, PFHFC, RHOFIL(1), RHOFIL(2), WOVENT,WHVENT,
269      1 DELTCP, TENV, PRFCOP, POWNOM
270      READ (IIN,5141) NFCOP, NFCSTB, PLSET1, PLSET2, VJANUL(1),
271      1 VJANUL(2),TKHXDI(1),TKHXDI(2)
272      READ (IIN,5130) FCVOLT, PRGRAT(1), PRGRAT(2), PRGTIM(1),
273      1 PRGTIM(2), PRGINT(1), PRGINT(2)
274      C
275      WRITE (IOT,6150) NFCOP  ,(L12(I, 1),I=1,4),MRFC,(L12(I, 2),I=1,4),
276      1 SRCFC   ,(L12(I, 3),I=1,4),ODTFC   ,(L12(I, 4),I=1,4),
277      2 SPWTFC  ,(L12(I, 5),I=1,4),TFCNOM(1),(L12(I, 6),I=1,4),
278      3 TFCNOM(2),(L12(I, 7),I=1,4),TF2IIN  ,(L12(I, 8),I=1,4),
279      4 TF2IOU  ,(L12(I, 9),I=1,4),TFOFC   ,(L12(I,10),I=1,4),
280      5 TFHFC   ,(L12(I,11),I=1,4),PFOFC   ,(L12(I,12),I=1,4),
281      6 PFHFC   ,(L12(I,13),I=1,4),RHOFIL(1),(L12(I,14),I=1,4),
282      7 RHOFIL(2),(L12(I,15),I=1,4),WOVENT  ,(L12(I,16),I=1,4),
283      8 WHVENT  ,(L12(I,17),I=1,4),DELTCP  ,(L12(I,18),I=1,4),
284      9 TENV    ,(L12(I,19),I=1,4),PRFCOP  ,(L12(I,20),I=1,4),
285      T POWNOM  ,(L12(I,21),I=1,4),PLSET1  ,(L12(I,22),I=1,4),
286      A PLSET2  ,(L12(I,23),I=1,4),VJANUL(1),(L12(I,24),I=1,4),
287      B VJANUL(2),(L12(I,25),I=1,4),TKHXDI(1),(L12(I,26),I=1,4),
288      C TKHXDI(2),(L12(I,27),I=1,4),FCVOLT  ,(L12(I,28),I=1,4),
289      D PRGRAT(1),(L12(I,29),I=1,4),PRGRAT(2),(L12(I,30),I=1,4),

```

```

***** .COMPIL *****
290 .E          PRGTIM(1),(L12(1,25),I=1,4),PRGTIM(2),(L12(1,26),I=1,4),
291 .F          PRGINT(1),(L12(1,27),I=1,4),PRGINT(2),(L12(1,28),I=1,4)
292 .C
293 .C
294 .C      113 CONTINUE
295 .C
296 .C      ***** INPUT TANK DATA
297 .C
298 .C      IF (PAGE(0)) WRITE (IOT,6050) (LO(1,12),I=1,9)
299 .C
300 .C      J=1
301 .C      READ (IIN,5060) (NOP (I,J),SATYPE(I ),SITYPE(I,J),SMTYPE(I,J),
302 .C      1 SPTYPE(I,J),SITEMP(I,J),SIPRES(I,J),SPGTEM(I,J),SOPRES(I,J),
303 .C      2 SVPRES(I,J),SHFLUX(I,J),SITHIK(I,J),FLDLOD(I ),SULGPC(I ),
304 .C      3 SMDIAM(I,J),SHOTEM(I,J),SHDELP(I,J),SPDELP(I,J),SGGTEM(I,J),
305 .C      4 SGGPC (I,J),SGMRAT(I,J),SNBAR (I ),I=1,2)
306 .C      WRITE (IOT,6060) (NOP (I,J),I=1,2),(L7(1,22),I=1,4),
307 .C      1 (SATYPE(I ),I=1,2),(L7(1,12),I=1,4),
308 .C      2 (SITYPE(I,J),I=1,2),(L7(1,10),I=1,4),
309 .C      3 (SMTYPE(I,J),I=1,2),(L7(1, 9),I=1,4),
310 .C      4 (SPTYPE(I,J),I=1,2),(L7(1,13),I=1,4)
311 .C
312 .C      WRITE (IOT,6090) (SITEMP(I,J),I=1,2),(L7(1, 6),I=1,4),
313 .C      1 (SIPRES(I,J),I=1,2),(L7(1, 7),I=1,4),
314 .C      2 (SPGTEM(I,J),I=1,2),(L7(1,15),I=1,4),
315 .C      3 (SOPRES(I,J),I=1,2),(L7(1,14),I=1,4),
316 .C      4 (SVPRES(I,J),I=1,2),(L7(1, 8),I=1,4),
317 .C      5 (SHFLUX(I,J),I=1,2),(L7(1, 5),I=1,4),
318 .C      6 (SITHIK(I,J),I=1,2),(L7(1,11),I=1,4),
319 .C      7 (FLDLOD(I ),I=1,2),(L7(1, 3),I=1,4),
320 .C      8 (SULGPC(I ),I=1,2),(L7(1, 4),I=1,4),
321 .C      9 (SMDIAM(I,J),I=1,2),(L7(1, 2),I=1,4),
322 .C      A (SHOTEM(I,J),I=1,2),(L7(1,18),I=1,4),
323 .C      B (SHDELP(I,J),I=1,2),(L7(1,16),I=1,4),
324 .C      C (SPDELP(I,J),I=1,2),(L7(1,17),I=1,4),
325 .C      D (SGGTEM(I,J),I=1,2),(L7(1,20),I=1,4),
326 .C      E (SGGPC (I,J),I=1,2),(L7(1,19),I=1,4),
327 .C      F (SGMRAT(I,J),I=1,2),(L7(1,21),I=1,4),
328 .C      G (SNBAR (I ),I=1,2),(L7(1,23),I=1,4)
329 .C
330 .C      READ (IIN,5062) IWOP,NOSHAP
331 .C
332 .C      WRITE (IOT,6091) IWOP, NOSHAP
333 .C
334 .C      CHECK FOR GENERAL TANK CONFIGURATION
335 .C      IF (IWOP .LT. 2 .OR. NOSHAP .EQ. 0) GO TO 210
336 .C      WRITE (IOT,6062) (LO(1,10),I=1,9)
337 .C      READ IN GENERAL TANK CONFIGURATION
338 .C      DO 200 I=1,NOSHAP
339 .C      READ (IIN,5062) JTKTYP(I),JFLTP(I),XD(I),YD(I),ZD(I)
340 .C      200 WRITE(IOT,6064) JTKTYP(I),JFLTP(I),XD(I),YD(I),ZD(I)
341 .C      210 CONTINUE
342 .C
343 .C      DO 2000 JSIM=1,5
344 .C
345 .C      IF (INBLK(SYSNUM,JSIM,SCRIT) .EQ. 0) GO TO 2000
346 .C      GO TO (1100,1200,1300,1400,1500),JSIM
347 .C

```



```

***** COMPIL *****
348 C ***** INPUT THE ACCUMULATOR DATA
349 C
350 1100 CONTINUE
351 IF(PAGE(0)) WRITE(6,6050) (LO(I,3),I=1,9)
352 C
353 READ (IIN,5030) (NAOP(I),AITYPE(I),AMTYPE(I),ATEMP(I),APRES(I),
354 1 AHFLUX(I),AITHIK(I),AVOL(I),ADIAM(I),ANDELP(I),
355 2 ANBAR(I),I=1,2)
356 C
357 WRITE (IOT,6060) (NAOP(I),I=1,2),(LT(I,22),I=1,4),
358 1 (AITYPE(I),I=1,2),(LT(I,10),I=1,4),
359 2 (AMTYPE(I),I=1,2),(LT(I,9),I=1,4)
360 C
361 WRITE (IOT,6090) (ATEMP(I),I=1,2),(L3(I,1),I=1,4),
362 1 (APRES(I),I=1,2),(LT(I,14),I=1,4),
363 2 (AHFLUX(I),I=1,2),(LT(I,5),I=1,4),
364 3 (AITHIK(I),I=1,2),(LT(I,11),I=1,4),
365 4 (AVOL(I),I=1,2),(L3(I,2),I=1,4),
366 5 (ADIAM(I),I=1,2),(LT(I,2),I=1,4),
367 6 (ANDELP(I),I=1,2),(L3(I,3),I=1,4),
368 7 (ANBAR(I),I=1,2),(LT(I,23),I=1,4)
369 GO TO 2000
370 C
371 C ***** INPUT THE HEX DATA
372 C
373 1200 CONTINUE
374 IF(PAGE(0)) WRITE(IOT,6050) (LO(I,4),I=1,9)
375 READ(IIN,5039) NUMHEX
376 READ(IIN,5040)((HEXHIT(I,J),HEXHOT(I,J),HEXCIT(I,J),HEXCOT(I,J),
377 1 HEXHIP(I,J),HEXHOP(I,J),HEXCIP(I,J),HEXCOP(I,J),
378 2 HXHDLP(I,J),HXCDLP(I,J),HXHRAT(I,J),HXCODE(I,J),
379 3 J=1,2),I=1,NUMHEX)
380 WRITE(IOT,6069) NUMHEX
381 WRITE(IOT,6051)
382 WRITE(IOT,6070)((HEXHIT(I,J),J=1,2),I=1,5),(L4(I,1),I=1,4),
383 1 ((HEXHOT(I,J),J=1,2),I=1,5),(L4(I,2),I=1,4),
384 2 ((HEXCIT(I,J),J=1,2),I=1,5),(L4(I,3),I=1,4),
385 3 ((HEXCOT(I,J),J=1,2),I=1,5),(L4(I,4),I=1,4),
386 4 ((HEXHIP(I,J),J=1,2),I=1,5),(L4(I,5),I=1,4),
387 5 ((HEXHOP(I,J),J=1,2),I=1,5),(L4(I,6),I=1,4),
388 6 ((HEXCIP(I,J),J=1,2),I=1,5),(L4(I,7),I=1,4),
389 7 ((HEXCOP(I,J),J=1,2),I=1,5),(L4(I,8),I=1,4),
390 8 ((HXHDLP(I,J),J=1,2),I=1,5),(L4(I,9),I=1,4),
391 9 ((HXCDLP(I,J),J=1,2),I=1,5),(L4(I,10),I=1,4),
392 10 ((HXHRAT(I,J),J=1,2),I=1,5),(L4(I,11),I=1,4)
393 IF (NUMHEX .LE. 5) GO TO 2000
394 WRITE (IOT,6052)
395 WRITE(IOT,6070)((HEXHIT(I,J),J=1,2),I=6,10),(L4(I,1),I=1,4),
396 1 ((HEXHOT(I,J),J=1,2),I=6,10),(L4(I,2),I=1,4),
397 2 ((HEXCIT(I,J),J=1,2),I=6,10),(L4(I,3),I=1,4),
398 3 ((HEXCOT(I,J),J=1,2),I=6,10),(L4(I,4),I=1,4),
399 4 ((HEXHIP(I,J),J=1,2),I=6,10),(L4(I,5),I=1,4),
400 5 ((HEXHOP(I,J),J=1,2),I=6,10),(L4(I,6),I=1,4),
401 6 ((HEXCIP(I,J),J=1,2),I=6,10),(L4(I,7),I=1,4),
402 7 ((HEXCOP(I,J),J=1,2),I=6,10),(L4(I,8),I=1,4),
403 8 ((HXHDLP(I,J),J=1,2),I=6,10),(L4(I,9),I=1,4),
404 9 ((HXCDLP(I,J),J=1,2),I=6,10),(L4(I,10),I=1,4),
405 10 ((HXHRAT(I,J),J=1,2),I=6,10),(L4(I,11),I=1,4)

```

```

***** COMPIL *****
406      GO TO 2000
407      C
408      ***** INPUT THE PUMP DATA
409      C
410      1300 CONTINUE
411      IF (PAGE(0)) WRITE(6,6050) (LO(I,5),I=1,9)
412      READ(5,5050) (PTYPE(I),PEFF(I),PNPSH(I),PSSPED(I),EPDELP(I),I=1,2),
413      1      (TPEFF(I),TPNPSH(I),TPCELP(I),TPWDOT(I),I=1,2),
414      2      (TEFF(I),TITEMP(I),TOTEMP(I),TMRATO(I),TGGPC(I),
415      3      I=1,2)
416      WRITE(6,6080) (PTYPE(I),I=1,2),(L5(I,1),I=1,4),
417      1      (PEFF(I),I=1,2),(L5(I,2),I=1,4),
418      2      (PNPSH(I),I=1,2),(L5(I,3),I=1,4),
419      3      (PSSPED(I),I=1,2),(L5(I,4),I=1,4),
420      4      (EPDELP(I),I=1,2),(L5(I,5),I=1,4),
421      5      (LO(I,1),I=1,9),
422      6      (TPEFF(I),I=1,2),(L6(I,1),I=1,4),
423      7      (TPNPSH(I),I=1,2),(L6(I,2),I=1,4),
424      8      (TPCELP(I),I=1,2),(L6(I,3),I=1,4),
425      9      (TPWDOT(I),I=1,2),(L6(I,4),I=1,4),
426      T      (LO(I,8),I=1,9),
427      1      (TEFF(I),I=1,2),(L9(I,1),I=1,4),
428      2      (TITEMP(I),I=1,2),(L9(I,2),I=1,4),
429      3      (TOTEMP(I),I=1,2),(L9(I,3),I=1,4),
430      4      (TMRATO(I),I=1,2),(L9(I,4),I=1,4),
431      5      (TGGPC(I),I=1,2),(L9(I,5),I=1,4)
432      GO TO 2000
433      C
434      ***** INPUT THE HEAT SOURCE DATA
435      C
436      1400 CONTINUE
437      IF (PAGE(0)) WRITE (IOT,6100) (LO(I,7),I=1,9)
438      READ (IIN,5039) NUMHSO
439      READ (IIN,5090) ((HSTYPE(I,J),HSMRAT(I,J),HSOTEM(I,J),HSAEE(I,J),
440      1      HSPRES(I,J),J=1,2),I=1,NUMHSO)
441      WRITE (IOT,6128) NUMHSO
442      WRITE (IOT,6129)
443      WRITE (IOT,6130) ((HSTYPE(I,J),J=1,2),I=1,5),(L10(I,1),I=1,4),
444      1      ((HSMRAT(I,J),J=1,2),I=1,5),(L10(I,2),I=1,4),
445      2      ((HSOTEM(I,J),J=1,2),I=1,5),(L10(I,3),I=1,4),
446      3      ((HSAEE(I,J),J=1,2),I=1,5),(L10(I,4),I=1,4),
447      4      ((HSPRES(I,J),J=1,2),I=1,5),(L10(I,5),I=1,4)
448      IF (NUMHSO .LE. 5) GO TO 2000
449      WRITE (IOT,6127)
450      WRITE (IOT,6130) ((HSTYPE(I,J),J=1,2),I=6,10),(L10(I,1),I=1,4),
451      1      ((HSMRAT(I,J),J=1,2),I=6,10),(L10(I,2),I=1,4),
452      2      ((HSOTEM(I,J),J=1,2),I=6,10),(L10(I,3),I=1,4),
453      3      ((HSAEE(I,J),J=1,2),I=6,10),(L10(I,4),I=1,4),
454      4      ((HSPRES(I,J),J=1,2),I=6,10),(L10(I,5),I=1,4)
455      GO TO 2000
456      C
457      ***** INPUT THE MOTOR DATA
458      C
459      1500 CONTINUE
460      IF (PAGE(0)) WRITE (IOT,6100) (LO(I,9),I=1,9)
461      READ(5,5100) MTYPE,MEFF,MSS,PDNSTY
462      WRITE (IOT,6140) MTYPE,MEFF,MSS,PDNSTY
463      C

```

***** COMPIL *****

464 2000 CONTINUE
465 C
466 RETURN
467 END

B-72

LMSC-A991396

PROCEDURE DEFINITION PROCESSOR - CONST

```
1  CONST* PROC
2  C
3      COMMON /CONST/ GRAVITY,PI,PI203
4  C
5  C
6  C      ***** CONST VARIABLE DEFINITION
7  C      *
8  C      * GRAVITY - UNIVERSAL GRAVITATIONAL CONSTANT
9  C      *
10 C
11 END
```

***** SUBROUTINE CONSUM

```

1      C
2      SUBROUTINE CONSUM
3      C
4      INCLUDE CCNTRL
5      C
6      GO TO (10,20), SYSNUM
7      C
8      10 CALL ENGINE
9      C
10     CALL FLORAT
11     C
12     GO TO 99
13     C
14     20 CALL APUFLO
15     C
16     CALL FLORAT
17     C
18     GO TO 99
19     C
20     99 CONTINUE
21     C
22     RETURN
23     END

```

MAIN DRIVER PROGRAM - CONTROL

```

1  C
2  LOGICAL PAGE
3  C
4  INCLUDE CCNTRL
5  INCLUDE CIGUNT
6  INCLUDE CKEYS
7  INCLUDE CPAGE
8  C
9  DATA NSPC / 'SUP' / ILST / 'LAS' /
10 C
11 5000 FORMAT(2A6,2X4,3XA3,1XA5/12A6)
12 5001 FORMAT(A3,A6,3X,A3,14X,A3,37X,1111)
13 6000 FORMAT ('0'19X '***' SYSTEM NAME STARTING 'A3.' IS IN ERROR '***')
14 6001 FORMAT ('0'/// 44X'*** YOU HAVE CALLED FOR THE SYSTEM 'A3,A6,
15 | '***')
16 C
17 C
18 C
19 CALL STODTA
20 CALL OTUNIT (IOT)
21 C
22 C
23 CALL DATE (9,DOR)
24 C
25 C
26 C
27 READ(5,5000) NAME,DEPT,BLD,EXT,CTITLE
28 NCASE = 1
29 INTGSY = 1
30 C
31 C
32 C
33 CALL INTAB
34 C
35 1 CONTINUE
36 KEY1 = 1
37 KEY2 = 0
38 IF (INTGSY.EQ.1) GO TO 5
39 READ (5,5000) NAME,DEPT,BLD,EXT,CTITLE
40 5 CONTINUE
41 READ (11N,5001) NSYS,NI,NCRIT,INTGR,MDTRC
42 DO 10 I=1,NBRYS
43 IF (NSYS.EQ. NMSYS(I)) GO TO 20
44 10 CONTINUE
45 WRITE (IOT,6000) NSYS
46 CALL EXIT
47 20 SYSNUM = 1
48 IF (PAGE(0)) WRITE (IOT,6001) NSYS,NI
49 SCRIT = 1
50 IF (NCRIT.EQ. NSPC) SCRIT = 2
51 INTGSY = 2
52 IF (INTGR.EQ. ILST) INTGSY = 1
53 C
54 C
55 C
56 CALL COMPIL
57 C

```

```

*****  CONTRL  *****
58      C          DO SYSTEM CALCULATIONS FOR THE GIVEN SYSTEM
59      C
60      C          CALL CRYCON
61      C
62      C          IF THIS IS AN INTEGRATED SYSTEM - GO READ NEXT
63      C          SYSTEM NAME
64      C          IF IT IS A LAST CARD EXIT
65      C
66      C          IF (INTGSY.EQ. 2) GO TO 1
67      C          CALL EXIT
68      C          END

```

PROCEDURE DEFINITION PROCESSOR - CPAGE

```

1  CPAGE* PROC
2  C
3      INTEGER DEPT ,BLD ,EXT ,PTITLE,CTITLE,OPTLUN,DOR ,TIME
4  C
5      COMMON /CPAGE/NAME(2),DEPT,BLD,EXT,MAXLIN,JNUM,PTITLE(6),
6      1 CTITLE(12),OPTLUN,NCASE,DOR(2),TIME(2)
7  C
8  C
9  C      ***** CPAGE VARIABLE DEFINITION
10 C
11 C      NAME  - USER'S NAME (INPUT).
12 C
13 C      DEPT  - USER'S DEPARTMENT (INPUT).
14 C
15 C      BLD   - USER'S BUILDING (INPUT).
16 C
17 C      EXT   - USER'S EXTENSION (INPUT).
18 C
19 C      MAXLIN - MAXIMUM LINES PER OUTPUT PAGE.
20 C
21 C      JNUM  - PROGRAM IDENTIFICATION NUMBER.
22 C
23 C      PTITLE - PROGRAM TITLE.
24 C
25 C      CTITLE - CASE TITLE (INPUT).
26 C
27 C      OPTLUN - OUTPUT LOGICAL UNIT NUMBER.
28 C
29 C      NCASE  - CASE NUMBER (INTERNALLY CREATED).
30 C
31 C      DOR   - DATE OF THIS RUN (INTERNALLY CREATED).
32 C
33 C      TIME  - TIME OF THIS RUN (INTERNALLY CREATED).
34 C
35      END

```

B-77

FUNCTION CPHI

```

1      FUNCTION CPHI(T)
2      COMMON /RFPR/ RF(10)
3      COMMON /CIGCP/G(9)
4      COMMON /SCRH/ X(40)
5      C
6      C.... ROUTINE TO CALCULATE INTEGRAL(CPO DT)
7      C
8      R=RF(5)
9      AK=RF(6)
10     T2=T*T
11     T3=T2*T
12     T4=T3*T
13     U=G(9)/T
14     X(1)=-1.0/(2.0*T2)
15     X(2)=-1.0/T
16     X(3)=ALOG(T)
17     X(4)=T
18     X(5)=T2/2.0
19     X(6)=T3/3.0
20     X(7)=T4/4.0
21     X(8)=U*T/(EXP(U)-1.0)
22     CPHI=0.0
23     DO 1 I=1,8
24     CPHI=CPHI+X(I)*G(I)
25     CPHI=CPHI*R
26     CPHI=CPHI*AK
27     RETURN
28     END

```

FUNCTION CPIG

```

1  FUNCTION CPIG(T)
2  .C
3  DOUBLE PRECISION G9, U, U2, DU
4  .C
5  COMMON /RFPR/ RF(10)
6  COMMON /CIGCP/G(9)
7  COMMON /SCRH/ X(40)
8  .C
9  C.... CALCULATE IDEAL GAS HEAT CAPACITY, CP
10 C
11 R=RF(5)
12 AK=RF(6)
13 T2=T*T
14 T3=T2*T
15 G9 = DBLE(G(9))
16 U = G9/DBLE(T)
17 U2 = U * U
18 X(1) = 1.0/T3
19 X(2) = 1.0/T2
20 X(3) = 1.0/T
21 X(4) = 1.0
22 X(5) = T
23 X(6) = T2
24 X(7) = T3
25 DU = DEXP(U)
26 X(8) = SNGL( U2 * DU/(DU-1.0D0)**2)
27 CPIG = 0.0
28 DO I = 1,8
29 CPIG=CPIG+X(I)*G(I)
30 CPIG=CPIG*R*AK
31 RETURN
32 END

```

FUNCTION CPSI

```

1      FUNCTION CPSI(T)
2      COMMON /CIGCP/ G(9)
3      COMMON /RFPR/ RF(10)
4      COMMON /SCRH/ X(40)
5      C
6      C.... ROUTINE TO CALCULATE INTEGRAL(CPO/T DT)
7      C
8      R=RF(5)
9      AK=RF(6)
10     T2=T*T
11     T3=T2*T
12     U=G(9)/T
13     EU = EXP(U)
14     X(1) = -1.0/(3.0*T3)
15     X(2) = -1.0/(2.0*T2)
16     X(3) = -1.0/T
17     X(4) = ALOG(T)
18     X(5) = T
19     X(6) = T2/2.0
20     X(7) = T3/3.0
21     X(8) = U/(EU-1.0) - ALOG(1.0-1.0/EU)
22     CPSI = 0.0
23     DO 1 I=1,8
24     CPSI=CPSI+X(I)*G(I)
25     CPSI=CPSI*R
26     CPSI=CPSI*AK
27     RETURN
28     END

```

***** PROCEDURE DEFINITION PROCESSOR - CPUMP

```

1  CPUMP* PROC
2  C
3  INTEGER PTYPE, PSTAGE
4  C
5  COMMON /CPUMP/ EPDELP(2),PEFF (2),PPDCH (2),PSSPED(2),PTYPE (2),
6  1  TPDELP(2),TPEFF (2),TPNPSH(2),TPWDT(2),TPWGT(2),PTEMP (2),
7  2  PPRES (2),PPWDT (2),PPDEL (2),PNPSH (2),PPRHO (2),PSTAGE(2),
8  3  PNPSPR(2),PMPEFF(2),PMVOL(2),PWEGHT(2),PMPOW (2),PSPD (2),
9  4  JJOPT (2),
10 5  LPP1(3,6), LPP2(4), LPP3(3,6), LPP4(4), LPP5(4)
11 C
12 DIMENSION UOTP1(12),UOTP2(12)
13 C
14 EQUIVALENCE (UOTP1 (5),PTEMP ),(UOTP1 ( 3),PPRES ),
15 1 (UOTP1 ( 5),PPWDT ),(UOTP1 ( 7),PPDEL ),(UOTP1 ( 9),PNPSH ),
16 2 (UOTP1 (11),PPRHO ),(UOTP2 ( 5),PNPSPR),(UOTP2 ( 3),PMPEFF),
17 3 (UOTP2 ( 5),PMVOL),(UOTP2 ( 7),PWEGHT),(UOTP2 ( 9),PMPOW ),
18 4 (UOTP2 (11),PSPD )
19 C
20 ***** CPUMP VARIABLE DEFINITION.
21 *
22 * PEFF -- PUMP EFFICIENCY.
23 *
24 * PMPEFF -- PUMP EFFICIENCY
25 *
26 * PMVOL -- PUMP VOLUME
27 *
28 * PMPOW -- PUMP POWER
29 *
30 * PWEGHT -- PUMP WEIGHT (LBS)
31 *
32 * PSPD -- PUMP SPEED
33 *
34 * PSTAGE -- NUMBER OF PUMP STAGES
35 *
36 * PNPSPR -- COMPUTED NPSH REQUIRED BY PUMP
37 *
38 * PTEMP -- TEMP OF FLUID TO PUMP
39 *
40 * PPRES -- PRESSURE OF FLUID FROM PUMP
41 *
42 * PPWDT -- FLOW RATE OF FLUID TO PUMP
43 *
44 * PPDEL -- PUMP DELTA-P
45 *
46 * PPRHO -- DENSITY OF FLUID TO PUMP
47 *
48 * JJOPT -- SELECTED OPTION FOR PUMP CAPABILITY, EQUALS
49 * 1 FOR MIN. POWER, EQUALS 2 FOR MIN. WEIGHT
50 *
51 * PTYPE -- PUMPTYPE = 1 FOR PUMP ONLY
52 * = 2 FOR TURBOPUMP ASSY.
53 *
54 * TPEFF -- TRANSFER PUMP EFFECIENCY
55 *
56 * TPNPSH -- TRANSFER PUMP NPSH
57 *

```

```

***** CPUMP *****
58 C      * TPWHT - TRANSFER PUMP WEIGHT
59 C      *
60 C      * TPDELP - TRANSFER PUMP DELTA-P
61 C      *
62 C      * TPWDOT - TRANSFER PUMP FLOW RATE
63 C      *
64 C      *
65 C      * PNPSH - PUMP NET POSITIVE SUCTION HEAD.
66 C      *
67 C      * PSSPED - PUMP SHAFT SPEED.
68 C      *
69 C      * EPDELP - ESTIMATED DELTA PRESSURE IN THE PUMP.
70 C      *
71 C      * PPDCH - PUMP DISCHARGE PRESSURE
72 C      *
73 C      * NOTE --- THE ABOVE VARIABLES ARE SPECIFIED FOR OXYGEN
74 C      *          IN THE FIRST WORD AND HYDROGEN IN THE SECOND
75 C      *          ***** WORD.
76 C
77 C      END

```

***** SUBROUTINE CPVTD

```

1      SUBROUTINE CPVTD(T,D,CP,CV)
2      COMMON /CRPR/ CR(3) /METH/ M
3      COMMON /RFPR/ RF(10)
4      C
5      C.... ROUTINE TO CALCULATE CV AND CP FROM THE EQUATION OF STATE
6      C
7      DC=CR(2)
8      TC=CR(3)
9      R =RF(5)
10     AK=RF(6)
11     CALL PFND(T,D,P)
12     IF(M.EQ.1)GO TO 1
13     IF(T.GT.TC)GO TO 1
14     IF(D.GT.DC)GO TO 2
15     CVO=CPIG(T)-(R*AK)
16     CV = CVO - (FING3(T,D)-FING3(T,0.0)) * AK
17     F1=T/D**2
18     F2=DPDT(T,D)
19     F3=DPDD(T,D)
20     CP=CV+(F1*F2**2/F3)*AK
21     RETURN
22     2  DT = 0.1
23     T1=T+DT
24     T2=T-DT
25     CALL LPROP(T1,P1,D,1,H,S,U1,Z)
26     CALL LPROP(T2,P2,D,1,H,S,U2,Z)
27     CV = (U1-U2)/(2.0*DT)
28     GO TO 3
29     END

```

***** SUBROUTINE CPVTD8

```
1      SUBROUTINE CPVTD8(TB,DB,CPB,CVB)
2      COMMON /RFPR/ RF(8)
3      WT=RF(7)
4      T = TB/1.8
5      D = DB * 453.59237E-3/(WT * 2.8316847E-2)
6      CALL CPVTD(T,D,CP,CV)
7      CPB = CP * 453.59237/(1.0543503E+3 * 1.8 * WT)
8      CVB = CV * 453.59237/(1.0543503E+3 * 1.8 * WT)
9      RETURN
10     END
```

***** SUBROUTINE CRYCON

```

1  C
2  SUBROUTINE CRYCON
3  C
4  LOGICAL DIAG,JP
5  C
6  INCLUDE :CNTRL
7  INCLUDE :CKEYS
8  C
9  KEY1 = 1613
10 C
11 K = 0
12 C
13 IF (SYSNUM.NE.2) GO TO 10
14 DO 6 I = 1,2
15 6 KSUBC(2,I+1) = JAPUS(SCRIT,I)
16 C
17 10 I = 1
18 K = 0
19 LREPT = 0
20 C
21 20 JKM = KSUBC(SYSNUM,I)
22 C
23 IF ZERO HAVE REACHED END OF CALLING SEQUENCE
24 IF (JKM.EQ. 0) GO TO 2200
25 C MDTRC = 15 DIAGNOSTIC TRACE SWITCH (INPUT IN CONTRL)
26 IF (MDTRC(JKM).EQ. 0) GO TO 50
27 C TURN ON DIAG. TRACE
28 JP = DIAG (-1,6HCRYCON)
29 C CALL ROUTINES DEPENDING ON SYSTEM SPECIFIED (SYSNUM)
30 50 GO TO (100,200,300,400,450,500,550,600,700,800,900,1000),JKM
31 C
32 100 CALL ACCRES
33 GO TO 2000
34 C
35 200 CALL ACQWT
36 GO TO 2000
37 C
38 300 GO TO (310,350),SCRIT
39 C
40 310 CALL APUSUB
41 GO TO 2000
42 C
43 350 CALL APUSUP
44 GO TO 2000
45 C
46 400 CALL CNPCAL
47 GO TO 2000
48 C
49 450 CALL FUELCL
50 GO TO 2000
51 C
52 500 CALL CONSUM
53 GO TO 2000
54 C
55 550 CALL ECLSS
56 GO TO 2000
57 C

```



```

***** CRYCON *****
58      600 CALL LIQRES
59      GO TO 2000
60      C
61      700 CALL TANK
62      GO TO 2000
63      C
64      800 K = K + 1
65      IF (SYSDUM .EQ. 2) K = 2
66      C
67      CALL TSIZE1 (K)
68      GO TO 2000
69      C
70      900 CALL WTACC
71      C
72      1000 CONTINUE
73      C
74      2000 CONTINUE
75      IF (LREPT) 10,2001,10
76      C
77      2001 IF (MDTRC(JK1)) .EQ. 0) GO TO 2100
78      C
79      C          TURN OFF DIAGNOSTIC TRACE SENTINEL
80      C
81      JP = DIAG (-2,6HCRYCON)
82      C
83      2100 I = I + 1
84      IF (I-9) 20,20,2200
85      C          END OF PROCESS THIS SYSTEM
86      2200 CONTINUE
87      C
88      C          PRINT COMPONENT WEIGHT SUMMARY
89      CALL OTPHSH
90      C
91      RETURN
92      END

```

***** FUNCTION CSPF21

```

1  C
2  FUNCTION CSPF21 (TMP)
3  C
4  DIMENSION T(9),CP(9)
5  C
6  DATA T / 415.0,465.0,510.0,545.0,575.0,590.0,610.0,660.0,700.0 /
7  DATA CP / .2335,.2380,.2455,.2530,.2630,.2700,.2800,.3160,.3520 /
8  C
9  DO 10 I=1,9
10 J = I
11 IF (T(J) .GE. TMP) GO TO 20
12 10 CONTINUE
13 C
14 20 IF (J .EQ. 1) J = 2
15 I = J - 1
16 CSPF21 = CP(I) + (TMP - T(I))*(CP(J) - CP(I))/(T(J) - T(I))
17 RETURN
18 END

```

***** SUBROUTINE CSUBP

```

1      SUBROUTINE CSUBP(TEMP,PRES,NGAS,C)
2      C
3      *****
4      C
5      NGAS = 1   FOR OXYGEN
6      C      = 2   FOR HYDROGEN
7      C      = 17  FOR HELIUM
8      C      = 18  FOR NITROGEN
9      C
10     *****
11     C
12     COMMON/CVALUE/ CPON, CVON, DON
13     C
14     T = TEMP
15     P = PRES
16     TTT = TEMP
17     PPP = PRES
18     C
19     GO TO (1,2,5,5,5,5,5,5,5,5,5,5,17,18),NGAS
20     C
21     COMPUTE CSUBP FOR OXYGEN
22     C
23     1 CALL CSUBPV(TTT,PPP,1)
24     C = CPON
25     RETURN
26     C
27     COMPUTE CSUBP FOR HYDROGEN
28     C
29     2 C = HPTCP(P,T)
30     RETURN
31     C
32     FOR ERRONEOUS NGAS INDEX
33     C
34     5 C = 0.0
35     WRITE (6,100)
36     100 FORMAT(10,20(1*),) ERRONEOUS NGAS INDEX WAS USED IN S.R. CSUBP
37     1 1,20(1*)/)
38     RETURN
39     C
40     COMPUTE CSUBP FOR HELIUM
41     C
42     17 TR = T/17.37
43     PR = P/41.82
44     IF (TR.GT.1.6460845) GO TO 22
45     POWER = 1.-4.62*ALOG10(TR)
46     GO TO 24
47     22 POWER=.5930735918-2.74*ALOG10(TR)
48     24 ACONST=10.**POWER
49     C = 1.252 + ACONST * PR * 386.3/1544.0
50     C
51     RETURN
52     C
53     COMPUTE CSUBP FOR NITROGEN
54     C
55     18 CALL CSUBPV(TTT,PPP,18)
56     C = CPON
57     RETURN

```

***** .CSUBP *****

58 C
59 END

***** SUBROUTINE CSUBPV

```

1      SUBROUTINE CSUBPV(T,P,NGAS)
2      C
3      COMMON /CRPR/ CR(3) /METH/ M
4      COMMON /RFPR/ RF(8)
5      COMMON/CVALUE/ CPON, CVON, DON
6      C
7      IF(NGAS.EQ.1) KF = 1
8      IF(NGAS.EQ.18) KF = 2
9      IF(KF.EQ.1)CALL DATA02
10     IF(KF.EQ.2)CALL DATAN2
11     C
12     KF =1 INDICATES OXYGEN
13     KF =2 INDICATES NITROGEN
14     C
15     M = 1
16     C
17     CALL DFNDB(T,P,D,ZI,0)
18     DON = D
19     C
20     CALL CPVTDB(T,D,CP,CV)
21     CPON = CP
22     CVON = CV
23     C
24     C
25     RETURN
26     C
27     END

```

***** SUBROUTINE CSUBPI

```

1      SUBROUTINE CSUBPI(T,R,C)
2
3      C COMPUTE SPECIFIC HEAT CAPACITY (IN BTU/LB-DEG R) AT CONSTANT PRESSURE
4      C FROM THE GRAPHICAL DATA OF THE COMBUSTION PRODUCTS OF H2 AND O2 AS
5      C A FUNCTION OF MIXTURE RATIO AND TURBINE INLET TEMPERATURE.
6
7      C T = TURBINE INLET TEMPERATURE
8      C R = WEIGHT RATIO OF O2 TO H2
9      C C = COMPUTED VALUE OF SPECIFIC HEAT OF EXHAUST GAS AT CONSTANT PRES.
10
11      DIMENSION X(9), CP(10,9), TT(10)
12
13      C VALUES OF MIXTURE RATIO FOR WHICH CP IS TABULATED.
14
15      DATA (X(I),I=1,9)/.5, .75, 1.0, 1.5, 2.0, 3.0, 4.0, 6.0, 8.0/
16
17      C VALUES OF TEMPERATURE FOR WHICH CP IS TABULATED.
18
19      DATA (TT(I), I=1,10)/ 1400.0, 1500.0, 1600.0, 1700.0, 1800.0,
20      I 1900.0, 2000.0, 2100.0, 2200.0, 2300.0/
21
22      C TABULATED VALUES OF CP AS A FUNCTION OF MIXTURE RATIO AND TEMPERATURE
23
24      DATA (CP(I,1),I=1,10)/ 2.40, 2.41, 2.425, 2.44, 2.46, 2.475, 2.495
25      I, 2.515, 2.53, 2.545/
26      DATA (CP(I,2),I=1,10)/ 2.07, 2.08, 2.1, 2.115, 2.13, 2.14, 2.155,
27      I 2.175, 2.19, 2.21/
28      DATA (CP(I,3),I=1,10)/ 1.83, 1.84, 1.855, 1.865, 1.88, 1.895, 1.91
29      I, 1.92, 1.935, 1.955/
30      DATA (CP(I,4),I=1,10)/ 1.485, 1.5, 1.51, 1.52, 1.535, 1.55, 1.56,
31      I 1.575, 1.59, 1.605/
32      DATA (CP(I,5),I=1,10)/ 1.25, 1.26, 1.275, 1.285, 1.295, 1.31,
33      I 1.325, 1.335, 1.345, 1.36/
34      DATA (CP(I,6),I=1,10)/ .965, .975, .985, .995, 1.005, 1.015, 1.025
35      I, 1.035, 1.05, 1.06/
36      DATA (CP(I,7),I=1,10)/ .8, .81, .82, .83, .84, .85, .86, .87, .88,
37      I .895/
38      DATA (CP(I,8),I=1,10)/ .605, .615, .625, .635, .645, .655, .665,
39      I .675, .685, .695/
40      DATA (CP(I,9),I=1,10)/ .5, .51, .52, .53, .54, .55, .56, .57, .58,
41      I .59/
42
43      C DETERMINE THE INTERVAL CONTAINING TURBINE INLET TEMPERATURE.
44
45      DO 100 II=2,10
46      I = II - 1
47      IF(T.LT. TT(II)) GO TO 150
48      100 CONTINUE
49
50      C DETERMINE THE INTERVAL CONTAINING THE MIXTURE RATIO.
51
52      DO 200 II=2,9
53      J = II - 1
54      IF(R.LT. X(II)) GO TO 220
55      200 CONTINUE
56
57      C INTERPOLATE FIRST ON TEMPERATURE AND THEN ON MIXTURE RATIO.

```

***** CSUBPI *****

```

58
59 C 220 C1 = ((CP(I+1,J) - CP(I,J))/(TT(I+1) - TT(I)))*(T - TT(I))
60      C1 = C1 + CP(I,J)
61      C2 = ((CP(I+1,J+1) - CP(I,J+1))/(TT(I+1) - TT(I)))*(T - TT(I))
62      C2 = C2 + CP(I,J+1)
63      C = ((C2 - C1)/(X(J+1) - X(J)))*(R - X(J)) + C1
64      RETURN
65      END

```

FUNCTION CSUBV

```

1      FUNCTION CSUBV(TEMP,PRES,NGAS)
2      C
3      C *****
4      C
5      C   NGAS = 1   FOR OXYGEN
6      C           = 2   FOR HYDROGEN
7      C           = 17  FOR HELIUM
8      C           = 18  FOR NITROGEN
9      C
10     C *****
11     C
12     C   COMMON/CVALUE/ CPON, CVON, DON
13     C
14     C   T = TEMP
15     C   P = PRES
16     C   TTT = TEMP
17     C   PPP = PRES
18     C
19     C   GO TO (1,2,5,5,5,5,5,5,5,5,5,5,5,5,5,17,18),NGAS
20     C
21     C   COMPUTE CSUBV FOR OXYGEN
22     C
23     C   1 CALL CSUBPV(TTT,PPP,1)
24     C   CSUBV = CVON
25     C   RETURN
26     C
27     C   COMPUTE CSUBV FOR HYDROGEN
28     C
29     C   2 CSUBV = HPTCV(P,T)
30     C   RETURN
31     C
32     C   FOR ERRONEOUS NGAS INDEX
33     C
34     C   5 CSUBV = 0.0
35     C   WRITE (6,100)
36     C   100 FORMAT(10,20(1*),1 ERRONEOUS NGAS INDEX WAS USED IN S.R. CSUBV
37     C   1 1,20(1*)/)
38     C   RETURN
39     C
40     C   COMPUTE CSUBV FOR HELIUM
41     C
42     C   17 CSUBV = 0.746
43     C   RETURN
44     C
45     C   COMPUTE CSUBV FOR NITROGEN
46     C
47     C   18 CALL CSUBPV(TTT,PPP,18)
48     C   CSUBV = CVON
49     C
50     C   RETURN
51     C
52     C   END

```


***** PROCEDURE DEFINITION PRØCESSØR - CSYSWT

```

1  CSYSWT* PROC
2  C
3  PARAMETER JWA=100, JWB=5
4  C
5  INTEGER CNUM
6  C
7  C
8  COMMON /CSYSWT/ CNUM, WTCOMP(JWA,JWB), WTCUMT(JWA,JWB),
9  CMPCOD(JWA,JWB), WTSURT, WTSUMT
10 C
11 C ***** CNUM - COMPONENT INDEX IN SERIAL LIST
12 C *
13 C * WTCOMP - COMPONENT WEIGHT
14 C *
15 C * WTCUMT - SYSTEM CUMULATIVE WEIGHT IN SERIAL LIST
16 C *
17 C * CMPCOD - COMPONENT ID CODE FROM CONFIGURATION
18 C *
19 C * WTSUBT - SUBSYSTEM WEIGHT SUBTOTAL
20 C *
21 C * WTSUMT - SYSTEM TOTAL WEIGHT
22 C *****
23 C
24 END

```

PROCEDURE DEFINITION PROCESSOR - CTAB

```

1  CTAB*  PROC
2  C
3  C      PARAMETER  NTBNI = 50 , NSBZ = 40
4  C
5  C      INTEGER TLA,TYPE
6  C
7  C      COMMON /CTAB/ TLA(NTBNI),NV,TYPE,NIP,ND,XTAB(NSBZ),YTAB(NSBZ),
8  C      I TAB(6,5),JTABID,NLTBL
9  C
10 C      DIMENSION ITAB(6,5)
11 C
12 C      EQUIVALENCE (ITAB,TAB)
13 C
14 C      ***** CTAB VARIABLE DEFINITION
15 C
16 C      TLA - TABLE LOCATION ARRAY
17 C      THIS ARRAY CONTAINS THE BEGINNING DRUM ADDRESS
18 C      - 1, FOR UP TO 50 TABLES
19 C
20 C      NV - NUMBER OF VALUES IN THE INPUT TABLE (NV<101).
21 C
22 C      TYPE - TYPE OF THE INPUT TABLE (0 = COEFFICIENT,
23 C              1 = DISCRETE ).
24 C
25 C      NIP - NUMBER OF TABLE VALUES TO BE USED IN
26 C      INTERPOLATION (NIP<NV).
27 C
28 C      ND - NUMBER OF DIMENSIONS FOR THE INPUT TABLE
29 C      (ND<7).
30 C
31 C      XTAB - ARRAY OF COEFFICIENTS FOR POLYNOMIAL EVALUATION
32 C      OR ARRAY OF VALUES OF THE INDEPENDENT VARIABLE
33 C      FOR INTERPOLATION.
34 C
35 C      YTAB - ARRAY OF VALUES OF THE DEPENDENT VARIABLE
36 C      FOR INTERPOLATION.
37 C
38 C      TAB - ARRAY OF VALUES OF THE REMAINING ND-2
39 C      ITAB INDEPENDENT VARIABLES FOR INTERPOLATION.
40 C      END

```

***** PROCEDURE DEFINITION PROCESSOR - CTABA

```
1 CTABA* PROC
2 C
3   PARAMETER MXWRD = 7000
4 C
5   COMMON /CTABA/ IDX1, TABLE(MXWRD)
6 C
7   DIMENSION ITABLE(MXWRD)
8 C
9   EQUIVALENCE (TABLE, ITABLE)
10 C
11  END
```

PROCEDURE DEFINITION PROCESSOR - CTANK

CTANK* PROC

C

PARAMETER IJA=2,IJB=1,IJC=30

C

INTEGER SATYPE,SITYPE,SMTYPE,SPTYPE

C

```
COMMON /CTANK/  FLDLOD(IJA  ),NOP  (IJA,IJB),SATYPE(IJA  ),
1 SGGPC (IJA,IJB),SGMRAT(IJA,IJB),SGOTEM(IJA,IJB),SHDELPA(IJA,IJB),
2 SHFLUX(IJA,IJB),SHOTEM(IJA,IJB),SHRATE(IJA,IJB),SIPRES(IJA,IJB),
3 SITEMP(IJA,IJB),SITYPE(IJA,IJB),SITHIK(IJA,IJB),SMDIAM(IJA,IJB),
4 SMTYPE(IJA,IJB),SOPRES(IJA,IJB),SPDELP(IJA,IJB),SPGTEM(IJA,IJB),
5 SPTYPE(IJA,IJB),SULGPC(IJA  ),SVFLD(IJA  ),SVOL  (IJA,IJB),
6 SVPRES(IJA,IJB),SVTOT(IJA,IJB),TIWT  (IJA,IJB),TSA  (IJA,IJB),
7 TWT  (IJA,IJB),WRPG  (IJA,IJB),WCPPG (IJA,IJB),WGGAPG(IJA,IJB),
8 SNBAR (IJA  ),WGGP  (IJA,IJB),WGGPPG(IJA,IJB),
9 WGGTOT(IJA  ),WGR  (IJA,IJB),WHETOT(IJA,IJB),WLR  (IJA  ),
A WLRT  (IJA,IJB),WMPG  (IJA,IJB),  WPGTOT(IJA,IJB),
B WPTOT (IJA  ),WTACQ (IJA,IJB),WTGGH2(IJA,IJB),WTGG02(IJA,IJB),
C WTHXPG(IJA  ),WTOTP (IJA  ),WTSVPG(IJA,IJB),WTTOT (IJA  ),
D  PRESHE(IJC,IJA,IJB),SVHT  (IJC,IJA,IJB),
E WDOTHE(IJC,IJA,IJB),WDOTPG(IJC,IJA,IJB)
F,LTZ1(3,14),INDXTK(IJA),LTZ2(2,3),LTZ3(3,4),TCYHT(IJA)
```

C

***** CTANK VARIABLE DEFINITION

C

* SATYPE - ACQUISITION TYPE

C

* SITYPE - INSULATION TYPE

C

* SMTYPE - MATERIAL TYPE

C

* SPTYPE - PRESURIZATION TYPE

C

* SMDIAM - MAXIMUM DIAMETER (FT)

C

* SHFLUX - HEAT FLUX (BTU/HR-FT**2)

C

* SITEMP - INITIAL TEMP (DEG. R)

C

* SIPRES - INITIAL PRES (PSIA)

C

* SVPRES - VENTING PRES (PSIA)

C

* SITHIK - INSULATION THICKNESS (IN)

C

* SOPRES - OPERATING PRES (PSIA)

C

* SPOTEM - PRESSURANT GAS TEMP (DEG. R)

C

* SHDELPA - HEX DELTA PRES (PSIA)

C

* SPDELP - PUMP DELTA PRES (PSIA)

C

* SHOTEM - HEX OUTLET TEMP (DEG. R)

C

* SGGPC - P SUB C OF THE GAS GEN. (PSIA)

C

* SGOTEM - GAS GEN. OUTLET TEMP (DEG. R)

C

```

***** CTANK *****
58 C *
59 C * SNBAR = NUMBER OF LAYERS OF INSULATION ON TANK
60 C *
61 C * WTOTP = TOTAL PROPELLANT IN TANK INITIALLY
62 C *
63 C * WLRT = LIQUID RESIDUALS COMPUTED IN TANK
64 C *
65 C ***** SGMRAT = GAS GEN. MIXTURE RATIO
66 C
67 END

```

PROCEDURE DEFINITION PROCESSOR - CTURBN

CTURBN* PROC

C

COMMON /CTURBN/ TEFF (2),TGGPC (2),TITEMP(2),THRATO(2),TOTEMP(2),
1 TTRMD (2),TWPTL(2),TWGTTR(2),TWMFNZ(2),TWNDCR(2),TWEGHT(2),
2 WGGGR(2),GWEGHT(2),WGTGGA(2),
3 LBN1(4),LTBN2(5,6)

C

DIMENSION UOTB1(12),UOTB2(6)

C

EQUIVALENCE (UOTB1 ,TTRMD),(UOTB1(3),TWPTL),
1 (UOTB1(5),TWGTTR),(UOTB1(7),TWMFNZ),(UOTB1(9),TWNDCR),
2 (UOTB1(11),TWEGHT),(UOTB2 ,WGGGR),(UOTB2(3),GWEGHT),
3 (UOTB2(5),WGTGGA)

C

***** CTURBN VARIABLE DEFINITION

C

* THRATO - TURBINE MIXTURE RATIO.

C

* TEFF - TURBINE EFFICIENCY.

C

* TITEMP - TURBINE INLET TEMPRATURE.

C

* TOTEMP - TURBINE OUTLET TEMPRATURE.

C

* TWEGHT - TURBINE HEIGHT

C

* GWEGHT - INTEGRATED GAS GEN. PROPELLANT WEIGHT

C

* WGGGR - GAS GENERATOR FLOW RATE

C

* WGTGGA - GAS GENERATOR WEIGHT

C

* TGGPC - TURBINE GAS GENERATOR CHAMBER PRESSURE

C

* TTRMD - TURBINE ROTOR MEAN DIAMETER

C

* TWPTL - WGT. POWER TRANSMISSION ELEMENT

C

* TWGTTR - WGT. TURBINE ROTOR

C

* TWMFNZ - WGT. INLET MANIFOLD AND NOZZLE

C

* TWNDCR - WGT. OF INDUCER ASSY.

C

* NOTE - VARIABLES ARE SPECIFIED FOR OXYGEN
IN THE FIRST WORD AND HYDROGEN IN
THE SECOND WORD

C

END

***** SUBROUTINE DATAN2

```

1      SUBROUTINE DATAN2.
2      COMMON /CEOS/G(41) /CVPN/GV(11) /CIGCP/GI(9) /CSL/CL(7) /CSV/CV(7)
3      .      /CRPR/CR(3) /CTEVP/CT(8) /RFP/RF(10) /METH/ M
4      C
5      C.... IF THE PROPERTIES OF NITROGEN ARE TO BE CALCULATED A CALL TO THIS
6      C.... SUBROUTINE MUST BE THE FIRST CALL STATEMENT IN THE MAIN PROGRAM
7      C
8      C.... THE COMMON BLOCKS INITIALIZED IN THIS ROUTINE HOLD THE FOLLOWING
9      C.... INFORMATION -
10     C
11     C      /CEOS/ G(41) COEFFICIENTS OF THE EQUATION OF STATE
12     C
13     C      /CIGCP/ GI(9) COEFFICIENTS OF THE IDEAL GAS HEAT CAPACITY EQUATION
14     C
15     C      /CVPN/ GV(11) COEFFICIENTS OF THE VAPOR PRESSURE EQUATION
16     C
17     C      /CRPR/ CR(3) THE CRITICAL PROPERTIES IN THE SAME UNITS AS THE
18     C      EQUATION OF STATE
19     C      CR(1)=CRITICAL PRESSURE
20     C      CR(2)=CRITICAL DENSITY
21     C      CR(3)=CRITICAL TEMPERATURE
22     C
23     C      /CSL/ SL(7) COEFFICIENTS OF EQUATION USED TO APPROXIMATE
24     C      THE SATURATED LIQUID DENSITY AS A FUNCTION OF
25     C      TEMPERATURE
26     C
27     C
28     C      /RFP/ RF(10) REFERENCE PROPERTIES
29     C      RF(1)=REFERENCE ENTHALPY AT TEMPERATURE TOM
30     C      RF(2)=REFERENCE ENTROPY AT TEMPERATURE TOS
31     C      RF(3)=TEMPERATURE TOM
32     C      RF(4)=TEMPERATURE TOS
33     C      RF(5)=GAS CONSTANT IN UNITS OF EQUATION OF STATE - R
34     C      RF(6)=CONVERSION FACTOR TO CHANGE UNITS OF
35     C      THE EQUATION OF STATE TO DESIRED ENERGY UNITS
36     C      RF(7)=MOLECULAR WEIGHT
37     C      RF(8)=TRIPLE POINT TEMPERATURE
38     C      RF(9) - NOT USED
39     C      RF(10) - NOT USED
40     C
41     C      /METH/ M INDICATES METHOD TO BE USED IN THE CALCULATION
42     C      OF LIQUID PROPERTIES
43     C      M=1 INDICATES ISOTHERM INTEGRATION THROUGH THE DOME
44     C      M=2 INDICATES THE USE OF THE CLAPEYRON EQUATION
45     C      THROUGH THE DOME
46     C
47     C      M=1
48     C      G( 1) = 0.13622477E-02
49     C      G( 2) = 0.10703247E+00
50     C      G( 3) = -0.24390072E+01
51     C      G( 4) = 0.34100745E+02
52     C      G( 5) = -0.42237431E+04
53     C      G( 6) = 0.10509860E-03
54     C      G( 7) = -0.11259483E-01
55     C      G( 8) = 0.14260079E-03
56     C      G( 9) = 0.18469850E+05
57     C      G(10) = 0.81114008E-07
58     C      G(11) = 0.23301165E-02

```

DATAN2

58	G(12) =	-0.50775258E+00
59	G(13) =	0.48502788E+04
60	G(14) =	-0.11365676E+02
61	G(15) =	-0.70743027E+00
62	G(16) =	0.75170665E+04
63	G(17) =	-0.11161412E+05
64	G(18) =	0.36879656E+03
65	G(19) =	-0.20131769E+05
66	G(20) =	-0.16971744E+05
67	G(21) =	-0.11971924E+06
68	G(22) =	-0.97521827E+02
69	G(23) =	0.55463971E+05
70	G(24) =	-0.17992045E+00
71	G(25) =	-0.25658293E+01
72	G(26) =	-0.41370772E+03
73	G(27) =	-0.25624542E+00
74	G(28) =	-0.12422237E+06
75	G(29) =	0.10355654E+04
76	G(30) =	-0.53369917E+09
77	G(31) =	-0.75741541E+08
78	G(32) =	0.58536717E+07
79	G(41) =	-0.56000000E+02
80	GV(1) =	0.83944094E+04
81	GV(2) =	-0.18900453E+04
82	GV(3) =	-0.72822292E+01
83	GV(4) =	0.00000000E+00
84	GV(5) =	0.55560638E+03
85	GV(6) =	-0.59445447E+05
86	GV(7) =	0.27154339E+07
87	GV(8) =	-0.48795359E+10
88	GV(9) =	0.50953608E+03
89	GV(10) =	0.10228510E+01
90	GV(11) =	0.19500000E+01
91	GI(1) =	-0.73521040E+03
92	GI(2) =	0.34223998E+02
93	GI(3) =	-0.55764828E+00
94	GI(4) =	0.35040423E+01
95	GI(5) =	-0.17339019E+04
96	GI(6) =	0.17465085E+07
97	GI(7) =	-0.35689203E+11
98	GI(8) =	0.10053872E+01
99	GI(9) =	0.33534061E+04
100	CL(1) =	0.19424403E+02
101	CL(2) =	0.57083748E+02
102	CL(3) =	-0.24326985E+03
103	CL(4) =	0.88516838E+03
104	CL(5) =	-0.16393680E+04
105	CL(6) =	0.11574320E+04
106	CL(7) =	0.10182210E+01
107	CV(1) =	0.16333345E+01
108	CV(2) =	-0.94043771E+01
109	CV(3) =	0.21852746E+02
110	CV(4) =	-0.10268744E+03
111	CV(5) =	0.18794974E+03
112	CV(6) =	-0.16402437E+03
113	CV(7) =	-0.95731639E+01
114	CR(1) =	0.33555000E+02
115	CR(2) =	0.11210000E+02


```

*****  DATAN2  *****
116      CR( 3) =    0.12620000E+03
117      CT( 1) =   -0.14206479E-02
118      CT( 2) =    0.12908809E-01
119      CT( 3) =    0.00
120      CT( 4) =    0.00
121      CT( 5) =    0.00
122      CT( 6) =    0.00
123      CT( 7) =    0.00
124      CT( 8) =    0.00
125      RF( 1) =    0.86690000E+04
126      RF( 2) =    0.19150200E+03
127      RF( 3) =    0.29815000E+03
128      RF( 4) =    0.29815000E+03
129      RF( 5) =    0.82053900E-01
130      RF( 6) =    0.10132780E+03
131      RF( 7) =    0.28013400E+02
132      RF( 8) =    0.63148000E+02
133      RETURN
134      END

```

***** SUBROUTINE DATA02

```

1      SUBROUTINE DATA02
2      COMMON /CEOS/G(41) /CVPN/GV(11) /CIGCP/IG(9) /CSL/SL(7) /CSV/SV(7)
3          /CRPR/CR(3) /CTEVP/CT(8) /RFRP/RF(10) /METH/ M
4
5      C
6      C.... IF THE PROPERTIES OF OXYGEN ARE TO BE CALCULATED A CALL TO THIS
7      C.... SUBROUTINE MUST BE THE FIRST CALL STATEMENT IN THE MAIN PROGRAM
8      C
9      C.... THE COMMON BLOCKS INITIALIZED IN THIS ROUTINE HOLD THE FOLLOWING
10     C.... INFORMATION -
11     C
12     C      /CEOS/ G(41) COEFFICIENTS OF THE EQUATION OF STATE
13     C
14     C      /CIGCP/ IG(9) COEFFICIENTS OF THE IDEAL GAS HEAT CAPACITY EQUATION
15     C
16     C      /CVPN/ GV(11) COEFFICIENTS OF THE VAPOR PRESSURE EQUATION
17     C
18     C      /CRPR/ CR(3) THE CRITICAL PROPERTIES IN THE SAME UNITS AS THE
19     C                     EQUATION OF STATE
20     C                     CR(1)=CRITICAL PRESSURE
21     C                     CR(2)=CRITICAL DENSITY
22     C                     CR(3)=CRITICAL TEMPERATURE
23     C
24     C      /CSL/ SL(7) COEFFICIENTS OF EQUATION USED TO APPROXIMATE
25     C                     THE SATURATED LIQUID DENSITY AS A FUNCTION OF
26     C                     TEMPERATURE
27     C
28     C      /CSV/ SV(7) COEFFICIENTS OF EQUATION USED TO APPROXIMATE
29     C                     THE SATURATED VAPOR DENSITY AS A FUNCTION
30     C                     OF THE TEMPERATURE
31     C
32     C      /RFRP/ RF(10) REFERENCE PROPERTIES
33     C                     RF(1)=REFERENCE ENTHALPY AT TEMPERATURE TOM
34     C                     RF(2)=REFERENCE ENTROPY AT TEMPERATURE TOS
35     C                     RF(3)=TEMPERATURE TOM
36     C                     RF(4)=TEMPERATURE TOS
37     C                     RF(5)=GAS CONSTANT IN UNITS OF EQUATION OF STATE - R
38     C                     RF(6)=CONVERSION FACTOR TO CHANGE UNITS OF
39     C                     THE EQUATION OF STATE TO DESIRED ENERGY UNITS
40     C                     RF(7)=MOLECULAR WEIGHT
41     C                     RF(8)=TRIPLE POINT TEMPERATURE
42     C                     RF(9) = NOT USED
43     C                     RF(10) = NOT USED
44     C
45     C      /METH/ M INDICATES METHOD TO BE USED IN THE CALCULATION
46     C                     OF LIQUID PROPERTIES
47     C                     M=1 INDICATES ISOTHERM INTEGRATION THROUGH THE DOME
48     C                     M=2 INDICATES THE USE OF THE CLAPEYRON EQUATION
49     C                     THROUGH THE DOME
50     C
51     C      M = 1
52     C      G( 1) = -0.43090454E-02
53     C      G( 2) =  0.35201737E+00
54     C      G( 3) = -0.58362214E+01
55     C      G( 4) =  0.24350909E+03
56     C      G( 5) = -0.12463612E+05
57     C      G( 6) =  0.12080882E-03
58     C      G( 7) = -0.55031700E-01
59     C      G( 8) = -0.10775786E-03

```

***** DATA02 *****

```

58      G( 9) =  0.27859971E+04
59      G(10) = -0.70406317E-05
60      G(11) =  0.73426722E-02
61      G(12) = -0.59386982E+00
62      G(13) = -0.63616842E-04
63      G(14) =  0.33034015E-03
64      G(15) = -0.85769299E-01
65      G(16) = -0.75461916E-05
66      G(17) =  0.99647836E-07
67      G(18) =  0.62468110E-04
68      G(19) = -0.73169850E-06
69      G(20) =  0.11734852E+04
70      G(21) = -0.39878104E+06
71      G(22) = -0.16682114E+02
72      G(23) =  0.10949860E+06
73      G(24) = -0.17847345E-01
74      G(25) =  0.18752562E+01
75      G(26) = -0.94101786E-04
76      G(27) = -0.57539682E+00
77      G(28) = -0.20715572E-07
78      G(29) =  0.15747135E-04
79      G(30) = -0.10757921E-09
80      G(31) = -0.70023860E-08
81      G(32) =  0.34354851E-07
82      G(41) = -0.56000000E-02
83      GV( 1) = -0.55819321E+03
84      GV( 2) = -0.12118871E+03
85      GV( 3) = -0.83456212E-01
86      GV( 4) =  0.0
87      GV( 5) =  0.16875024E-04
88      GV( 6) = -0.21262477E-06
89      GV( 7) =  0.95741097E-09
90      GV( 8) = -0.16617640E-11
91      GV( 9) =  0.27545606E+02
92      GV(10) =  0.26603644E-02
93      GV(11) =  0.19157553E+01
94      GI( 1) = -0.49819985E+04
95      GI( 2) =  0.23024778E+03
96      GI( 3) = -0.34556532E+01
97      GI( 4) =  0.35218767E+01
98      GI( 5) = -0.43542022E-04
99      GI( 6) =  0.13463534E-07
100     GI( 7) =  0.16205983E-10
101     GI( 8) =  0.10314685E+01
102     GI( 9) =  0.22391811E+04
103     CL( 1) =  0.23418919E+02
104     CL( 2) =  0.63188646E+02
105     CL( 3) = -0.17790822E+03
106     CL( 4) =  0.41295084E+03
107     CL( 5) = -0.50136455E+03
108     CL( 6) =  0.24255815E+03
109     CL( 7) =  0.11440740E+01
110     CV( 1) =  0.18784168E+01
111     CV( 2) = -0.96025698E+01
112     CV( 3) =  0.21060675E+02
113     CV( 4) = -0.88962992E+02
114     CV( 5) =  0.14637955E+03
115     CV( 6) = -0.12348431E+03

```

***** DATA02 *****

```
116      CV( 7) = -0.83291153E-01
117      CR( 1) =  0.49770000E+02
118      CR( 2) =  0.13630000E+02
119      CR( 3) =  0.15458100E+03
120      CT( 1) = -0.11152354E-02
121      CT( 2) =  0.11160348E-01
122      CT( 3) = -0.72432619E-04
123      CT( 4) =  0.88579161E-05
124      CT( 5) = -0.56777226E-06
125      CT( 6) =  0.18657256E-07
126      CT( 7) = -0.29899729E-09
127      CT( 8) =  0.18530363E-11
128      RF( 1) =  0.86820000E+04
129      RF( 2) =  0.20503700E+03
130      RF( 3) =  0.29815000E+03
131      RF( 4) =  0.29815000E+03
132      RF( 5) =  0.08205390E+00
133      RF( 6) =  0.10132780E+03
134      RF( 7) =  0.31998800E+02
135      RF( 8) =  0.54351300E+02
136      RETURN
137      END
```

SUBROUTINE DCALC

```

1      SUBROUTINE DCALC(D,T,P,DL,DH)
2      DATA MAX/30/
3      DATA EPS/7.5E-9/
4      C
5      C.... ROUTINE TO PERFORM ITERATIVE SOLUTION OF THE EQUATION OF STATE
6      C
7      C.... DL IS LOWER BOUND ON DENSITY
8      C.... DH IS UPPER BOUND ON DENSITY
9      C
10     C.... THE DESIRED DENSITY MUST LIE BETWEEN DL AND DH
11     C
12     C.... ALGORITHM IS MODIFIED VERSION OF
13     C.... 1A QUADRATIC FORMULA FOR FINDING THE ROOT OF AN EQUATION BY
14     C.... L. G. CHAMBERS MATHEMATICS OF COMPUTATION VOL 25 NO 114 APRIL (1971)
15     C
16     IDS=0
17     IC=0
18     2 CONTINUE
19     D1=DL
20     D2=DH
21     CALL PFND(T,D1,P1)
22     CALL PFND(T,D2,P2)
23     Y1=P1-P
24     Y2=P2-P
25     1 IC=IC+1
26     IF(IC.GT.MAX)GO TO 5
27     IF(IDS.EQ.1)GO TO 7
28     IF(IC.GT.20)GO TO 6
29     DS=(D1*Y2-D2*Y1)/(Y2-Y1)
30     GO TO 8
31     7 DS = (D1+D2)/2.0
32     8 CONTINUE
33     CALL PFND(T,DS,PS)
34     YS=PS-P
35     D3=DS*Y1*Y2/((YS-Y2)*(YS-Y1)) +
36     . D1*YS*Y2/((Y1-Y2)*(Y1-YS)) +
37     . D2*YS*Y1/((Y2-Y1)*(Y2-YS))
38     CALL PFND(T,D3,P3)
39     Y3=P3-P
40     IF(DABS(Y3).LE.EPS)GO TO 3
41     IF(Y3.GT.0.0) GO TO 12
42     IF(Y3.LT.Y1)GO TO 12
43     Y1=Y3
44     D1=D3
45     12 IF(YS.GT.0.0) GO TO 13
46     IF(YS.LT.Y1)GO TO 13
47     Y1=YS
48     D1=DS
49     13 IF(Y3.LT.0.0) GO TO 14
50     IF(Y3.GT.Y2)GO TO 14
51     Y2=Y3
52     D2=D3
53     14 IF(YS.LT.0.0) GO TO 1
54     IF(YS.GT.Y2)GO TO 1
55     Y2=YS
56     D2=DS
57     GO TO 1

```

***** DCALC *****

```

58      3 CONTINUE
59      D=D3
60      RETURN
61      5 WRITE(6,300)T,P,DL,DH,D1,Y1,D2,Y2,DS,Y3,D3,Y3
62      300 FORMAT(1,'*** DCALC FAILED TO CONVERGE ***',/,
63      .          .          .          .          .          .          .          .          .          .
64      .          .          .          .          .          .          .          .          .          .
65      .          .          .          .          .          .          .          .          .          .
66      .          .          .          .          .          .          .          .          .          .
67      .          .          .          .          .          .          .          .          .          .
68      .          .          .          .          .          .          .          .          .          .
69      .          .          .          .          .          .          .          .          .          .
70      .          .          .          .          .          .          .          .          .          .
71      .          .          .          .          .          .          .          .          .          .
72      .          .          .          .          .          .          .          .          .          .
73      .          .          .          .          .          .          .          .          .          .
74      .          .          .          .          .          .          .          .          .          .
75      D=DS
76      RETURN
77      6 CONTINUE
78      IDS=1
79      IC=0
80      GO TO 2
81      END

```

SUBROUTINE DENSON

```

1      SUBROUTINE DENSON(TEMP,PRES,NGAS,DENS,ZEE)
2      C
3      COMMON /METH/ M
4      C
5      C      INPUT TO THIS SUBROUTINE MUST BE IN BRITISH UNITS
6      C
7      T = TEMP
8      P = PRES
9      M = 1
10     C
11     IF(NGAS.EQ.1) KF = 1
12     IF(NGAS.EQ.18) KF = 2
13     C
14     C      KF = 1 CALL IN OXYGEN PARAMETERS
15     C      KF = 2 CALL IN NITROGEN PARAMETERS
16     C
17     IF(KF.EQ.1) CALL DATAO2
18     IF(KF.EQ.2) CALL DATAN2
19     C
20     CALL DFNDB(T,P,D,Z,D)
21     DENS = D
22     ZEE = Z
23     C
24     RETURN
25     END

```

SUBROUTINE DFND

```

1      SUBROUTINE DFND(T,P,D,Z,K)
2      COMMON /PFPR/ RF(10)
3      COMMON /CRPR/ CP(3)
4      C
5      C.... ROUTINE TO GENERATE TRAIL DENSITIES FOR ITERATIVE SOLUTION OF
6      C.... THE EQUATION OF STATE FOR DENSITY GIVEN TEMPERATURE AND PRESSURE
7      C
8      C      K = 0 INDICATES SINGLE PHASE POINT
9      C      K = 1 INDICATES T + P ARE FOR THE SATURATED LIQUID
10     C      K = 2 INDICATES T + P ARE FOR THE SATURATED VAPOR
11     C
12     IF((K.LE.2).AND.(K.GE.0))GO TO 1
13     WRITE(6,300)K
14     300 FORMAT('*** ERROR IN CALL DFND ***',/,
15             '      K MUST EQUAL 0, 1, OR 2',/,
16             '      K = ',I10)
17     RETURN
18     1 PC=CP(1)
19     DC=CR(2)
20     TC=CR(3)
21     R = RF(5)
22     IF(K.GT.0)GO TO 5
23     IF(T.GR.TC)GO TO 2
24     VP=VPH(T)
25     IF(P.LE.VP)GO TO 3
26     4 DH = 3.1 * DC
27     DL=DSATL(T)
28     CALL DCALC(D,T,P,DL,DH)
29     Z = P/(D*R*T)
30     RETURN
31     3 DL = 0.0
32     DH=DSATV(T)
33     CALL DCALC(D,T,P,DL,DH)
34     Z = P/(D*R*T)
35     RETURN
36     2 DL = 0.0
37     DH = 3.1 * DC
38     IF((T.GT.1000.0).AND.(P.LT.300.0)) DH=DC
39     CALL DCALC(D,T,P,DL,DH)
40     Z = P/(D*R*T)
41     RETURN
42     5 IF(K.EQ.1)GO TO 4
43     GO TO 3
44     END

```


SUBROUTINE DFNDB

```
1      SUBROUTINE DFNDB(TB,PB,DB,ZB,K)
2      COMMON /RFPR/ RF(10)
3      WTE=RF(7)
4      T = TB/1.8
5      P = PB * 6.8947572E+9/1.01325E+5
6      CALL DFND(T,P,D,Z,K)
7      DB = D * WT * 2.8316847E-2/453.59237E-9
8      ZB = Z
9      RETURN
10     END
```

***** FUNCTION DIAG

```

1      C      * * * * *
2      C      * ROUTINE NAME = DIAGNOSTIC TRACE ROUTINE *
3      C      * ROUTINE LANG = FORTRAN V UNIVAC 1108 EXEC 24 *
4      C      * PROGRAMMER  = R. BOLLINGER 1943 102 26933 *
5      C      * DATE CODED   = APRIL 6, 1970 *
6      C      * * * * *
7      C
8      C      FUNCTION DIAG(I,NAME)
9      C
10     C      ***** EXPLANATION OF THE CALLING SEQUENCE
11     C      1 - PRINT FLAG
12     C      IF I = -2, THE DIAGNOSTIC TRACE WILL BE TURNED OFF.
13     C      IF I = -1, THE DIAGNOSTIC TRACE WILL BE TURNED ON.
14     C      IF I = 0, DIAG WILL PRINT THE XXXXXX ENTRED MESSAGE.
15     C      IF I = 1, DIAG WILL PRINT THE XXXXXX EXITED MESSAGE.
16     C      IF I = 2, DIAG WILL PRINT THE XXXXXX INTRNL MESSAGE.
17     C
18     C      NAME = THE NAME OF THE ROUTINE TO BE PRINTED OUT WITH
19     C      I = 0,1,2.
20     C
21     C      LOGICAL PAGE,DIAG
22     C
23     C      DIMENSION MSG(3)
24     C
25     C      DATA (MSG(J),J=1,3)/'ENTRED','EXITED','INTRNL'/
26     C
27     C      DIAG = .FALSE.
28     C      IF(I)10,40,40
29     C      10 IF(I + 1) 20,30,30
30     C      20 IFLG = 0
31     C      RETURN
32     C      30 IFLG = 1
33     C      IF(PAGE(0)) WRITE(6,6000)
34     C      RETURN
35     C      40 IF(IFLG.EQ.0) RETURN
36     C      WRITE(6,6010) NAME,MSG(I+1)
37     C      DIAG = .TRUE.
38     C      RETURN
39     C      6000 FORMAT('0**DIAGNOSTIC TRACE**')
40     C
41     C      6010 FORMAT(1XA6,1XA6)
42     C      END

```

FUNCTION DPDD

```

1      FUNCTION DPDD(T,D)
2      COMMON /RFPR/ RF(10)
3      COMMON /CEOS/G(41)
4      COMMON /SCRH/ X(40)
5      C
6      C.... CALCULATES DP/DD OF THE EQUATION OF STATE
7      C
8      R=RF(5)
9      D2=D*D
10     D3=D2*D
11     D4=D3*D
12     D5=D4*D
13     D6=D5*D
14     D7=D6*D
15     D8=D7*D
16     D9=D8*D
17     D10=D9*D
18     D11=D10*D
19     D12=D11*D
20     D13=D12*D
21     TS= SQRT(T)
22     T2=T*T
23     T3=T2*T
24     T4=T3*T
25     GM=G(41)
26     F= EXP(GM*D2)
27     F1=2.00*F*GM*D
28     F21=3.000*F*D2 +F1*D3
29     F22=5.000*F*D4 +F1*D5
30     F23=7.000*F*D6 +F1*D7
31     F24=9.000*F*D8 +F1*D9
32     F25=11.00*F*D10+F1*D11
33     F26=13.00*F*D12+F1*D13
34     X( 1)=2.00*D*T
35     X( 2)=2.00*D*TS
36     X( 3)=2.00*D
37     X( 4)=2.00*D/T
38     X( 5)=2.00*D/T2
39     X( 6)=3.00*D2*T
40     X( 7)=3.00*D2
41     X( 8)=3.00*D2/T
42     X( 9)=3.00*D2/T2
43     X(10)=4.00*D3*T
44     X(11)=4.00*D3
45     X(12)=4.00*D3/T
46     X(13)=5.00*D4
47     X(14)=6.00*D5/T
48     X(15)=6.00*D5/T2
49     X(16)=7.00*D6/T
50     X(17)=8.00*D7/T
51     X(18)=8.00*D7/T2
52     X(19)=9.00*D8/T2
53     X(20)=F21/T2
54     X(21)=F21/T3
55     X(22)=F22/T2
56     X(23)=F22/T4
57     X(24)=F23/T2

```

```

*****      DPDD      *****
58          X(25)=F23/T3
59          X(26)=F24/T2
60          X(27)=F24/T4
61          X(28)=F25/T2
62          X(29)=F25/T3
63          X(30)=F26/T2
64          X(31)=F26/T3
65          X(32)=F26/T4
66          N=32
67          DPDD=0.0
68          DO 1 K=1,N
69      1    DPDD=DPDD+X(K)*G(K)
70          DPDD=DPDD+R*T
71          RETURN
72          END

```

***** FUNCTION DPDD8

```

1      FUNCTION DPDD8(TB,DB)
2      COMMON /RFPR/ RF(10)
3      WT=RF(7)
4      T  = TB/1.8
5      D  = DB * 453.59237E-3/(WT * 2.8316847E-2)
6      DPDD8 = DPDD(T,D) * 453.59237E-3 * 1.01325E+5/(WT * 6.8947572E+9 *
7      | 2.8316847E-2)
8      RETURN
9      END

```

***** FUNCTION DPDT

```

1      FUNCTION DPDT(T,D)
2      COMMON /CEOS/G(41)
3      COMMON /RFPR/ RF(10)
4      COMMON /SCRH/ X(40)
5      C
6      C.... CALCULATES DP/DT OF THE EQUATION OF STATE
7      C
8      R=RF(5)
9      D2=D*D
10     D3=D2*D
11     D4=D3*D
12     D5=D4*D
13     D6=D5*D
14     D7=D6*D
15     D8=D7*D
16     D9=D8*D
17     D10=D9*D
18     D11=D10*D
19     D12=D11*D
20     D13=D12*D
21     TS= SQRT(T)
22     T2=T*T
23     T3=T2*T
24     T4=T3*T
25     T5=T4*T
26     GM=G(41)
27     F= EXP(GM*D2)
28     X( 1)=D2
29     X( 2)=D2/(2.00*TS)
30     X( 3)=0.0
31     X( 4)=-D2/T2
32     X( 5)=-2.00*D2/T3
33     X( 6)=D3
34     X( 7)=0.0
35     X( 8)=-D3/T2
36     X( 9)=-2.00*D3/T3
37     X(10)=D4
38     X(11)=0.0
39     X(12)=-D4/T2
40     X(13)=0.0
41     X(14)=-D6/T2
42     X(15)=-2.00*D6/T3
43     X(16)=-D7/T2
44     X(17)=-D8/T2
45     X(18)=-2.00*D8/T3
46     X(19)=-2.00*D9/T3
47     X(20)=-2.00*D3*F/T3
48     X(21)=-3.00*D3*F/T4
49     X(22)=-2.00*D5*F/T3
50     X(23)=-4.00*D5*F/T5
51     X(24)=-2.00*D7*F/T3
52     X(25)=-3.00*D7*F/T4
53     X(26)=-2.00*D9*F/T3
54     X(27)=-4.00*D9*F/T5
55     X(28)=-2.00*D11*F/T3
56     X(29)=-3.00*D11*F/T4
57     X(30)=-2.00*D13*F/T3

```

```

***** DPDT *****
58      X(31)=-3.00*D13*F/T4
59      X(32)=-4.00*D13*F/T5
60      N=32
61      DPDT=0.0
62      DO I K=1,N
63      DPDT=DPDT+X(K)*G(K)
64      DPDT=DPDT+R*D
65      RETURN
66      END

```

***** FUNCTION DPDTB

```
1      FUNCTION DPDTB(TB,DB)
2      COMMON /RFPR/ RF(10)
3      WT=RF(7)
4      T  = TB/1.8
5      D  = DB * 453.59237E-3/(WT * 2.8316847E-2)
6      DPDTB = DPDT(T,D) * 1.01325E+5/(1.8 * 6.8947572E+3)
7      RETURN
8      END
```


***** SUBROUTINE DPDTVP

```

1      SUBROUTINE DPDTVP(T,P,DPDT)
2      COMMON/CVPM/G(11) /CRPR/CR(3)
3      COMMON /SCRH/ X(40)
4      C
5      C.... CALCULATE DP/DT FOR THE VAPOR PRESSURE EQUATION
6      C
7      TC=CR(3)
8      A=G(11)
9      T2=T*T
10     T3=T*T2
11     T4=T*T3
12     T5=T*T4
13     X(1) = -1.0/T2
14     X(2) = 0.0
15     X(3) = 1.0
16     X(4) = 2.0*T
17     X(5) = 3.0*T2
18     X(6) = 4.0*T3
19     X(7) = 5.0*T4
20     X(8) = 6.0*T5
21     X(9) = 1.0/T
22     X(10) = (TC-T)**(A-1.0)*(-A)
23     DPDT = 0.0
24     DO 1 I=1,10
25     1 DPDT=DPDT+X(I)*G(I)
26     DPDT=DPDT*P
27     RETURN
28     END

```

***** FUNCTION DSATL

```

1      FUNCTION DSATL(T)
2      COMMON /CRPR/CR(3) /CSL/6(7)
3      COMMON /SCRH/ B(40)
4      C
5      C.... THIS FUNCTION SUPPLIES AN APPROXIMATE VALUE FOR THE
6      C.... DENSITY OF THE SATURATED LIQUID
7      C
8      TC=CR(3)
9      X=(TC-T)/TC
10     X2=X*X
11     X3=X*X2
12     X4=X*X3
13     X5=X*X4
14     B(1) = 1.0
15     B(2)=X
16     B(3)=X2
17     B(4)=X3
18     B(5)=X4
19     B(6)=X5
20     B(7) = ALOG(X)
21     DSL = 0.0
22     DO 1 I=1,7
23     1 DSL=DSL+B(I)*6(I)
24     DSATL=DSL
25     RETURN
26     END

```

***** FUNCTION DSATV

```

1      FUNCTION DSATV(T)
2      COMMON /CRPR/CR(3) /CSV/G(7)
3      COMMON /SCRH/ B(40)
4      C
5      C.... THIS FUNCTION SUPPLIES AN APPROXIMATE VALUE FOR THE
6      C.... DENSITY OF THE SATURATED VAPOR
7      C
8      TC=CR(3)
9      X=(TC-T)/TC
10     X2=X*X
11     X3=X*X2
12     X4=X*X3
13     X5=X*X4
14     B(1) = 1.0
15     B(2)=X
16     B(3)=X2
17     B(4)=X3
18     B(5)=X4
19     B(6)=X5
20     B(7) = ALOG(X)
21     DSV = 0.0
22     DO 1 I=1,7
23     1 DSV=DSV+B(I)*G(I)
24     DSV = EXP(DSV)
25     DSATV=DSV
26     RETURN
27     .END

```

***** PROCEDURE DEFINITION PRØCESSØR - DUMMY

```
1 DUMMY* PROC.  
2 C  
3 DIMENSION SCRATCH(100)  
4 C  
5 COMMON /CDUMMY/ IGRBAG(100)  
6 C  
7 EQUIVALENCE (IGRBAG,SCRATCH)  
8 C  
9 C ***** CDUMMY VARIABLE DESCRIPTION  
10 C  
11 C IGRBAG - A SCRATCH AREA FOR VARIABLE STORAGE WHICH IS  
12 C NOT TO BE SAVE FROM ON ROUTINE TO ANOTHER.  
13 END
```

***** SUBROUTINE ECLSS

```

1      SUBROUTINE ECLSS
2      C
3      LOGICAL JP,PAGE
4      C
5      C
6      INCLUDE CACCUH
7      INCLUDE CAPU
8      INCLUDE CDCYCL
9      INCLUDE CECLSS
10     INCLUDE CENG
11     INCLUDE CFUEL
12     INCLUDE CHEX
13     INCLUDE CHSARC
14     INCLUDE CIOUNT
15     INCLUDE CMATRL
16     INCLUDE CONST
17     INCLUDE CPUMP
18     INCLUDE CTANK
19     INCLUDE TABLOK
20     C
21     DIMENSION WDOTX(MHX,2)
22     EQUIVALENCE (WDOTCF,WDOTX)
23     C
24     C *****
25     C
26     JP = PAGE(0)
27     C
28     WRITE (IOT,6000)
29     6000 FORMAT(/T38,'*** INITIATE PROGRAM AND CHARACTERIZE LIFE SUPPORT P
30     IARAMETERS ***'/)
31     WRITE (IOT,6001)
32     6001 FORMAT(//T2'PARAMETER',T14,'CYCLE-1',T26,'CYCLE-2',
33     1 T38,'CYCLE-3',T50,'CYCLE-4',T62,'CYCLE-5',T74,'CYCLE-6',T86,
34     2 'CYCLE-7',T98,'CYCLE-8',T110,'CYCLE-9',T121,'CYCLE-10'/)
35     C
36     C *****
37     C *****
38     C
39     C BEGIN COMPUTATIONS FOR SUPERCRITICAL STORAGE
40     C
41     C *****
42     C
43     C *** SET SELECTED INPUT PARAMETERS FOR CONVENIENCE
44     C
45     POP02 = SOPRES(1,1)
46     POPN2 = SOPRES(2,1)
47     PVP02 = SVPRES(1,1)
48     PVPN2 = SVPRES(2,1)
49     T02IN = SITEMP(1,1)
50     TN2IN = SITEMP(2,1)
51     C
52     C *** COMPUTE QUANTITY OF FLUIDS CONSUMED FOR LIFE SUPPORT,
53     C *** LEAKAGE, AND AIRLOCK OR CABIN REPRESSURIZATION, COMPUTE
54     C *** GAS REQD. FOR REPRESSURIZATION AND TOTAL GAS REQHTS.
55     C
56     O2MCON = 0.0
57     O2LCON = 0.0

```

B-122

LMSC-A991396

```

***** ECLSS *****
58      N2LCON = 0.0
59      I = 0
60      DO 10 I1 = 1, NDCYCL, 2
61      I = I + 1
62      O2MWT(I) = (O2FNOM/24.0) * NCREW * DCYCLE(I1)
63      O2MCON = O2MCON + O2MWT(I)      B LBS
64      O2LWT(I) = 0.21 * (GLKRAT/24.0) * DCYCLE(I1)
65      O2LCON = O2LCON + O2LWT(I)      B LBS
66      N2LWT(I) = 0.79 * (GLKRAT/24.0) * DCYCLE(I1)
67      N2LCON = N2LCON + N2LWT(I)      B LBS
68      C
69      C 10 CONTINUE
70      C
71      GASWGT = CABVOL/13.2743      B AIR AT 14.7 PSIA AND 70 F
72      O2REPR = 0.21 * NRPRES * GASWGT
73      N2REPR = 0.79 * NRPRES * GASWGT
74      O2CONS = O2MCON + O2LCON + O2REPR      B LBS
75      N2CONS = N2LCON + N2REPR      B LBS
76      C
77      C *****
78      C
79      C
80      C *** OUTPUT THE DATA COMPUTED TO THIS POINT ***
81      C
82      WRITE (IOT, 6002)
83      6002 FORMAT(/T3, 'COMPUTE QUANTITY OF FLUIDS CONSUMED EACH INTERVAL AND
84      IFOP ENTIRE MISSION')
85      WRITE (IOT, 600)
86      600 FORMAT(/)
87      WRITE (IOT, 601) (O2MWT(N), N=1, KCYCLE)
88      601 FORMAT(T3, 'O2MWT =', T13, 10(F8.5, 4X))
89      WRITE (IOT, 602) (O2LWT(N), N=1, KCYCLE)
90      602 FORMAT(T3, 'O2LWT =', T13, 10(F8.5, 4X))
91      WRITE (IOT, 603) (N2LWT(N), N=1, KCYCLE)
92      603 FORMAT(T3, 'N2LWT =', T13, 10(F8.5, 4X))
93      WRITE (IOT, 604) O2MCON, O2LCON
94      604 FORMAT(T3, 'O2MCON =', T13, F8.3, T25, 'O2LCON =', T37, F8.3)
95      WRITE (IOT, 605) N2LCON, GASWGT
96      605 FORMAT(T3, 'N2LCON =', T13, F8.3, T25, 'GASWGT =', T37, F8.3)
97      WRITE (IOT, 606) O2REPR, N2REPR
98      606 FORMAT(T3, 'O2REPR =', T13, F8.3, T25, 'N2REPR =', T37, F8.3)
99      WRITE (IOT, 607) O2CONS, N2CONS
100     607 FORMAT(T3, 'O2CONS =', T13, F8.3, T25, 'N2CONS =', T37, F8.3)
101     C
102     C *** COMPUTE THE CONTINGENCY RESERVE GASES REQD. FOR THE
103     C *** MISSION
104     C
105     O2MRES = (O2FNOM/24.0) * NCREW * NDARES * 24.0
106     O2LRES = 0.21 * (GLKRAT/24.0) * NDARES * 24.0
107     N2LRES = 0.79 * (GLKRAT/24.0) * NDARES * 24.0
108     O2RES = O2MRES + O2LRES      B LBS
109     N2RES = N2LRES      B LBS
110     C
111     C *** COMPUTE USEABLE GAS CONSUMABLES TOTAL
112     C
113     O2TOTU = O2CONS + O2RES      B LBS
114     N2TOTU = N2CONS + N2RES      B LBS
115     C

```

```

***** ECLSS *****
116 C *****
117 C
118 C *** OUTPUT THE DATA COMPUTED TO THIS POINT ***
119 C
120 WRITE (IOT,6003)
121 6003 FORMAT(/T3,'CONTINGENCY RESERVE FLUIDS REQD.FOR MISSION')
122 WRITE (IOT,600)
123 WRITE (IOT,608) O2MRES, O2LRES, N2LRES
124 608 FORMAT(T3,'O2MRES =',T13,F8.3,T25,'O2LRES =',T37,F8.3,T50,'N2LRES
125 I=',T60,F8.3)
126 WRITE (IOT,609) O2RES, N2RES
127 609 FORMAT(T3,'O2RES =',T13,F8.3,T25,'N2RES =',T37,F8.3)
128 C
129 C
130 WRITE (IOT,6004)
131 6004 FORMAT(/T3,'TOTAL USABLE FLUIDS AVAILABLE')
132 WRITE (IOT,600)
133 WRITE (IOT,610) O2TOTU, N2TOTU
134 610 FORMAT(T3,'O2TOTU =',T13,F8.3,T25,'N2TOTU =',T37,F8.3)
135 C
136 C *** COMPUTE NOMINAL FLOWRATE AND REPRESSURIZATION FLOWRATE
137 C *** FOR GASES, COMPUTE MAX FLOW RATE FOR GASES.
138 C *** COMPUTE QUANTITY GASES CONSUMED EACH INTERVAL
139 C
140 C
141 I = 0
142 DO 15 II = 1,NDCYCL,2
143 I = I + 1
144 WDOTON(I) = (O2MWT(I) + O2LWT(I))/DCYCLE(I)      0 LBS PER HR
145 WDOTNN(I) = N2LWT(I)/DCYCLE(I)                  0 LBS PER HR
146 IF(RPRTIM(I).EQ.0.0) GO TO 18
147 WDOTOR(I) = (O2REPR/NRPRES)/RPRTIM(I)            0 LBS PER HR
148 WDOTNR(I) = (N2REPR/NRPRES)/RPRTIM(I)            0 LBS PER HR
149 18 WDT02(I) = WDOTON(I) + WDOTOR(I)
150 WDTN2(I) = WDOTNN(I) + WDOTNR(I)                  0 LBS PER HR
151 WDT02(I) = WDOTON(I)*DCYCLE(I) + WDOTOR(I)*RPRTIM(I)  0 LBS
152 WDTN2(I) = WDOTNN(I)*DCYCLE(I) + WDOTNR(I)*RPRTIM(I)  0 LBS
153 15 CONTINUE
154 C
155 WDTOMX = 0.0
156 WDTNMX = 0.0
157 DO 17 I = 1,KCYCLE
158 IF(WDT02(I).LT.WDTOMX) GO TO 16
159 WDTOMX = WDT02(I)
160 IMAX = I
161 IF(WDTN2(I).LT.WDTNMX) GO TO 17
162 WDTNMX = WDTN2(I)
163 JMAX = I
164 17 CONTINUE
165 C
166 WDOTI(1) = WDTOMX/3600.0      0 LBS PER SECOND
167 WDOTI(2) = WDTNMX/3600.0      0 LBS PER SECOND
168 WDOTT(1) = WDOTI(1)           0 LBS PER SECOND
169 WDOTT(2) = WDOTI(2)           0 LBS PER SECOND
170 C
171 C *****
172 C
173 C *** OUTPUT THE DATA COMPUTED TO THIS POINT ***

```

```

***** ECLSS *****
174 C
175 WRITE (IOT,6005)
176 6005 FORMAT(/T3,'NOMINAL FLOWRATE, REPRESSURIZATION FLOWRATE, MAX-FLOWR-
177 IATE, AND QUANTITY EACH FLUID CONSUMED PER INTERVAL')
178 WRITE (IOT,600)
179 WRITE (IOT,611) (WDOTON(N),N=1,KCYCLE)
180 611 FORMAT(T3,'WDOTON =',T13,10(F8.5,4X))
181 WRITE (IOT,612) (WDOTNN(N),N=1,KCYCLE)
182 612 FORMAT(T3,'WDOTNN =',T13,10(F8.5,4X))
183 WRITE (IOT,613) (WDOTOR(N),N=1,KCYCLE)
184 613 FORMAT(T3,'WDOTOR =',T13,10(F8.5,4X))
185 WRITE (IOT,614) (WDOTNR(N),N=1,KCYCLE)
186 614 FORMAT(T3,'WDOTNR =',T13,10(F8.5,4X))
187 WRITE (IOT,615) (WDTOR(N),N=1,KCYCLE)
188 615 FORMAT(T3,'WDTOR =',T13,10(F8.5,4X))
189 WRITE (IOT,616) (WDTN2(N),N=1,KCYCLE)
190 616 FORMAT(T3,'WDTN2 =',T13,10(F8.5,4X))
191 WRITE (IOT,617) WDTOMX, WDTNMX
192 617 FORMAT(T3,'WDTOMX =',T13,F8.3,T25,'WDTNMX =',T13,F8.3)
193 WRITE (IOT,619) (WTO2(N),N=1,KCYCLE)
194 619 FORMAT(T3,'WTO2 =',T13,10(F8.4,4X))
195 WRITE (IOT,620) (WTN2(N),N=1,KCYCLE)
196 620 FORMAT(T3,'WTN2 =',T13,10(F8.4,4X))
197 C
198 C
199 C
200 CALL FINTAB(NTBID(8))
201 XTAB(1) = POPO2
202 XTAB(2) = RHOBEG(1)
203 TEMPO2 = MIPE(2,XTAB)
204 C
205 CALL FINTAB(NTBID(41))
206 XTAB(1) = POPN2
207 XTAB(2) = RHOBEG(2)
208 TEMPN2 = MIPE(2,XTAB)
209 C
210 C
211 C
212 C
213 C
214 C
215 C
216 C
217 C
218 C
219 C
220 C
221 C
222 C
223 C
224 C
225 C
226 C
227 C
228 C
229 C
230 C
231 C

```

*** DETERMINE INITIAL FLUID TANK TEMPERATURES
 *** DETERMINE INITIAL CSUBV VALUES FOR TANK T AND P CONDITIONS
 *** COMPUTE FLUID COMPRESSIBILITY AT FINAL TANK CONDITIONS
 *** COMPUTE THE PERCENT OF USABLE FLUIDS WITHDRAWN
 *** EACH INTERVAL

```

TKOW = 0.0
TKNW = 0.0
QDTOMX = 0.0
QDTNMX = 0.0
HWTOMX = 0.0
HWTNMX = 0.0
HWTCTT = 0.0
HWTNTT = 0.0
TWTOMX = 0.0

```



```

***** ECL55 *****
232 TWTNMH = 0.0
233 TWTOTT = 0.0
234 TWTNTT = 0.0
235 C
236 DO 20 I = 1, KCYCLE
237 TKOW = TKOW + WTO2(I)
238 TKO2DP(I) = TKOW
239 TKNW = TKNW + WTN2(I)
240 TKN2DP(I) = TKNW
241 C
242 PCOXWD(I) = TKO2DP(I)/O2TOTU
243 PCN2WD(I) = TKN2DP(I)/N2TOTU
244 C
245 C *** COMPUTE DENSITY OF FLUIDS AS FUNCTION OF PERCENT WITHDRAWN
246 C
247 C1 = 144.0/(RHOBEG(1) * (1544.2546/31.9988))
248 C2 = 144.0/(RHOBEG(2) * (1544.2546/28.0134))
249 C
250 C3 = 1.0 - ((C1 * TKFPRS(1))/(ZFO * TKFTEM(1)))
251 C4 = 1.0 - ((C2 * TKFPRS(2))/(ZFN * TKFTEM(2)))
252 C
253 O2RHO(I) = RHOBEG(1) * (1.0 - (PCOXWD(I) * C3))
254 N2RHO(I) = RHOBEG(2) * (1.0 - (PCN2WD(I) * C4))
255 C
256 C *** COMPUTE FLUID TEMPERATURE IN TANKS FOR EACH INTERVAL
257 C
258 CALL FINTAB (NTRID(8))
259 XTAB(1) = POPO2
260 XTAB(2) = O2RHO(I)
261 O2TEMP(I) = MIPE(2,XTAB)
262 C
263 CALL FINTAB (NTRID(41))
264 XTAB(1) = POPN2
265 XTAB(2) = N2RHO(I)
266 N2TEMP(I) = MIPE(2,XTAB)
267 C
268 C *** COMPUTE SPECIFIC HEAT INPUT (DG/DH) FOR FLUIDS AS F(D,P)
269 C *** COMPUTE ENERGY DERIVATIVE (DP/DO) FOR FLUIDS AS F(D,P)
270 C
271 CALL PHTON(O2TEMP(I),O2RHO(I),1,PHI,THETA)
272 DGDH02(I) = THETA
273 DPDU02(I) = PHI
274 C
275 CALL PHTON(N2TEMP(I),N2RHO(I),18,PHI,THETA)
276 DGDH02(I) = THETA
277 DPDU02(I) = PHI
278 C
279 C *** SIZE CONDITIONING HEAT EXCHANGERS FOR FLUIDS
280 C
281 HLSO = OXENTH(POPO2,TLNOM(1))
282 HLSN = NIENH(POPN2,TLNOM(2))
283 O2H(I) = OXENTH(POPO2,O2TEMP(I))
284 N2H(I) = NIENH(POPN2,N2TEMP(I))
285 C
286 QDTH(I) = WDT02(I) * (HLSO - O2H(I))
287 QDTHX = AMAX1(QDTHX,QDTH(I))
288 QDTHN(I) = WDTN2(I) * (HLSN - N2H(I))
289 QDTHNX = AMAX1(QDTHNX,QDTHN(I))

```

```

***** ECLSS *****
290 C
291 C *** COMPUTE POWER REQUIRED TO PROVIDE ENERGY IN HEAT XCHNGR.
292 C
293 HWAT02(I) = GDTOR(I) * 0.293
294 HWTOMX = AMAXI(HWTOMX,HWAT02(I))
295 HWTOTT = HWTOTT + HWAT02(I)
296 HWATN2(I) = GDTNR(I) * 0.293
297 HWTNMX = AMAXI(HWTNMX,HWATN2(I))
298 HWTNTT = HWTNTT + HWATN2(I)
299 C
300 C *** SIZE O2 TANK AND N2 TANK HEAT REQUIREMENTS
301 C
302 GDTTKO(I) = WDT02(I) * DQDMO2(I)
303 GDTTKN(I) = WDTN2(I) * DQDMN2(I)
304 C
305 C CALCULATE THE TANK HEATER RATINGS FOR EACH FLUID TANK BASED UPON
306 C INPUT HEATER DIAMETER AND LENGTH
307 C * PROGRAM WILL USE CALCULATED VALUES ONLY IF INPUT HTRFLX(2)=0.0 *
308 C
309 HTRPA1 = (TWTOMX/0.293)/(PI * HTRDIA(1) * HTRLNG(1))
310 HTRPA2 = (TWTNMX/0.293)/(PI * HTRDIA(2) * HTRLNG(2))
311 C
312 C *** COMPUTE POWER REQUIRED TO PROVIDE ENERGY INTO TANKS
313 C
314 TWAT02(I) = GDTTKO(I) * 0.293
315 TWTOMX = AMAXI(TWTOMX,TWAT02(I))
316 TWTOTT = TWTOTT + TWAT02(I)
317 TWATN2(I) = GDTTKN(I) * 0.293
318 TWTNMX = AMAXI(TWTNMX,TWATN2(I))
319 TWTNTT = TWTNTT + TWATN2(I)
320 20 CONTINUE
321 C
322 TOTWMX = HWTOMX + HWTNMX + TWTOMX + TWTNMX
323 C
324 TOTWAT = HWTOTT + HWTNTT + TWTOTT + TWTNTT
325 C
326 TOTPOW = TOTWAT/1000.0
327 C
328 C *****
329 C
330 C *** OUTPUT THE DATA COMPUTED TO THIS POINT ***
331 C
332 JP = PAGE(0)
333 WRITE (IOT,6116)
334 6116 FORMAT(//T42, '*** CONTINUE COMPUTATION OF - ECLSS - PARAMETERS ***'
335 1//)
336 WRITE (IOT,6001)
337 WRITE (IOT,6006)
338 6006 FORMAT(/T3, 'DETERMINE FLUID TANK CONDITIONS FOR DUTY CYCLE')
339 WRITE (IOT,600)
340 WRITE (IOT,621) TEMPO2, TEMPN2
341 621 FORMAT(T3, 'TEMPO2 =', T13, F8.2, T25, 'TEMPN2 =', T13, F8.2)
342 WRITE (IOT,622) CISBVC, CISBVN, ZFO, ZFN
343 622 FORMAT(T3, 'CISBVC =', T13, F8.4, T25, 'CISBVN =', T13, F8.4, T50, 'ZFO'
344 1=' ', T63, F8.4, T73, 'ZFN =', T86, F8.4)
345 WRITE (IOT,623) (PCOXWD(N), N=1, KCYCLE)
346 623 FORMAT(T3, 'PCOXWD =', T13, 10(F8.6, 4X))
347 WRITE (IOT,624) (PCN2WD(N), N=1, KCYCLE)

```

***** ECLSS *****

```

348      624 FORMAT(T3,'PCN2WD =',T13,10(F8.6,4X))
349      WRITE (IOT,625) (O2RHO(N),N=1,KCYCLE)
350      625 FORMAT(T3,'O2RHO =',T13,10(F8.4,4X))
351      WRITE (IOT,626) (N2RHO(N),N=1,KCYCLE)
352      626 FORMAT(T3,'N2RHO =',T13,10(F8.4,4X))
353      WRITE (IOT,627) (O2TEMP(N),N=1,KCYCLE)
354      627 FORMAT(T3,'O2TEMP =',T13,10(F8.2,4X))
355      WRITE (IOT,628) (N2TEMP(N),N=1,KCYCLE)
356      628 FORMAT(T3,'N2TEMP =',T13,10(F8.2,4X))
357      WRITE (IOT,629) (DDDMO2(N),N=1,KCYCLE)
358      629 FORMAT(T3,'DDDMO2 =',T13,10(F8.2,4X))
359      WRITE (IOT,630) (DDDMN2(N),N=1,KCYCLE)
360      630 FORMAT(T3,'DDDMN2 =',T13,10(F8.2,4X))
361      WRITE (IOT,631) (DPDUO2(N),N=1,KCYCLE)
362      631 FORMAT(T3,'DPDUO2 =',T13,10(F8.3,4X))
363      WRITE (IOT,632) (DPDUN2(N),N=1,KCYCLE)
364      632 FORMAT(T3,'DPDUN2 =',T13,10(F8.3,4X))
365
366      C
367      6007 FORMAT(/T3,'SIZE FLUID CONDITIONING HEAT EXCHANGERS')
368      WRITE (IOT,600)
369      WRITE (IOT,618) HLSO, HLSN
370      618 FORMAT(T3,'HLSO =',T13,F8.3,T25,'HLSN =',T37,F8.3)
371      WRITE (IOT,633) (O2H(N),N=1,KCYCLE)
372      633 FORMAT(T3,'O2H =',T13,10(F8.3,4X))
373      WRITE (IOT,634) (N2H(N),N=1,KCYCLE)
374      634 FORMAT(T3,'N2H =',T13,10(F8.3,4X))
375      WRITE (IOT,635) (QDTOR(N),N=1,KCYCLE)
376      635 FORMAT(T3,'QDTOR =',T13,10(F8.1,4X))
377      WRITE (IOT,636) (QDTNR(N),N=1,KCYCLE)
378      636 FORMAT(T3,'QDTNR =',T13,10(F8.1,4X))
379      WRITE (IOT,637) QDTOMX, QDTNMX
380      637 FORMAT(T3,'QDTOMX =',T13,F8.1,T25,'QDTNMX =',T37,F8.1)
381      WRITE (IOT,638) (HWATO2(N),N=1,KCYCLE)
382      638 FORMAT(T3,'HWATO2 =',T13,10(F8.2,4X))
383      WRITE (IOT,639) (HWATN2(N),N=1,KCYCLE)
384      639 FORMAT(T3,'HWATN2 =',T13,10(F8.2,4X))
385      WRITE (IOT,640) HWTOMX, HWTNMX
386      640 FORMAT(T3,'HWTOMX =',T13,F8.2,T25,'HWTNMX =',T37,F8.2)
387      WRITE (IOT,641) HWTOTT, HWTNTT
388      641 FORMAT(T3,'HWTOTT =',T13,F8.1,T25,'HWTNTT =',T37,F8.1)
389      WRITE (IOT,6008)
390      6008 FORMAT(/T3,'SIZE FLUID TANK HEAT REQUIREMENTS')
391      WRITE (IOT,600)
392      WRITE (IOT,642) (QDTTKO(N),N=1,KCYCLE)
393      642 FORMAT(T3,'QDTTKO =',T13,10(F8.1,4X))
394      WRITE (IOT,643) (QDTTKN(N),N=1,KCYCLE)
395      643 FORMAT(T3,'QDTTKN =',T13,10(F8.1,4X))
396      WRITE (IOT,644) (TWATO2(N),N=1,KCYCLE)
397      644 FORMAT(T3,'TWATO2 =',T13,10(F8.1,4X))
398      WRITE (IOT,645) (TWATN2(N),N=1,KCYCLE)
399      645 FORMAT(T3,'TWATN2 =',T13,10(F8.1,4X))
400      WRITE (IOT,646) TWTOMX, TWTNMX
401      646 FORMAT(T3,'TWTOMX =',T13,F8.1,T25,'TWTNMX =',T37,F8.1)
402      WRITE (IOT,670) HTRRA1, HTRRA2
403      670 FORMAT(T3,'O2 TANK HEATER RATING=',F6.1,T35,'N2 TANK HEATER RATING
404      =',F6.1)
405
      C

```

```

***** ECLSS *****
406 C      *** COMPUTE FLUID DENSITIES AT FINAL TEMP AND PRES CONDITIONS
407 C
408       TKOMXT = 0.0
409       TKNMXT = 0.0
410       DO 30 I = 1, KCYCLE
411         TKOMXT = AMAX1(TKOMXT, O2TEMP(I))
412         TKNMXT = AMAX1(TKNMXT, N2TEMP(I))
413       30 CONTINUE
414 C
415       CALL DENSON(TKOMXT, TKFPRS(1), 1, RHOEND(1), ZEO)
416       CALL DENSON(TKNMXT, TKFPRS(2), 2, RHOEND(2), ZEN)
417 C
418       *** COMPUTE WEIGHT OF RESIDUAL FLUIDS IN TANKS
419 C
420       WTRSID(1) = (RHOEND(1)/RHOBE(1))*(1.0-(RHOEND(1)/RHOBE(1)))
421       I * (O2CONS + O2RES)
422       WTRSID(2) = (RHOEND(2)/RHOBE(2))*(1.0-(RHOEND(2)/RHOBE(2)))
423       I * (N2CONS + N2RES)
424 C
425       *** COMPUTE VOLUME OF THE FLUID TANKS
426 C
427       VOLTK(1) = (O2TOTU + WTRSID(1))/(0.97*(RHOBE(1) - RHOEND(1)))
428       VOLTK(2) = (N2TOTU + WTRSID(2))/(0.97*(RHOBE(2) - RHOEND(2)))
429 C
430       *** COMPUTE AREA OF SPHERICAL FLUID TANKS
431 C
432       ARETK(1) = 4.84 * (VOLTK(1)**0.667)
433       ARETK(2) = 4.84 * (VOLTK(2)**0.667)
434 C
435       *** COMPUTE HEAT LEAK INTO FLUID TANKS
436 C
437       QLKOTK = 0.0
438       QLKNTK = 0.0
439       I = 0
440       DO 40 I = 2, NDCYCL, 2
441         I = I + 1
442         CALL TCOND(TENVR, O2TEMP(I), SNBAR(1), SITHIK(1, I), SITYPE(1, I), TNCND)
443         QO2LK(I) = TNCND * ARETK(1) * DCYCLE(I)
444         QLKOTK = QLKOTK + QO2LK(I)
445 C
446         CALL TCOND(TENVR, N2TEMP(I), SNBAR(2), SITHIK(2, I), SITYPE(2, I), TNCND)
447         QN2LK(I) = TNCND * ARETK(2) * DCYCLE(I)
448         QLKNTK = QLKNTK + QN2LK(I)
449 C
450       40 CONTINUE
451 C
452       *** COMPUTE QUANTITY OF FLUIDS VENTED DURING MISSION INTERVALS
453       *** AND TOTAL FLUIDS VENTED DURING THE MISSION
454 C
455       WVO2 = 0.0
456       WVN2 = 0.0
457       PVNTO = SVPRES(1, 1)
458       PVNTN = SVPRES(2, 1)
459       DO 50 I = 1, KCYCLE
460         CSBV02 = CSUBV(O2TEMP(I), POPO2, 1)
461         CSBVN2 = CSUBV(N2TEMP(I), POPN2, 1)
462         QREGDO(I) = ((VOLTK(1)*CSBV02)/48.9) * (PVNTO - POPO2) * 144.0
463         QREGDN(I) = ((VOLTK(2)*CSBVN2)/54.9) * (PVNTN - POPN2) * 144.0

```

```

***** ECLSS *****
464      IF(GREQDO(I).GT.G02LK(I)) GO TO 45
465      DELQ = G02LK(I) - GREQDO(I)
466      WTVNTO(I) = DELQ/(CSBV02 * 02TEMP(I) * ((PVNTO/POPO2) - 1.0))
467      45 WVO2 = WVO2 + WTVNTO(I)
468      IF(GREQDN(I).GT.GN2LK(I)) GO TO 47
469      DELGN = GN2LK(I) - GREQDN(I)
470      WTVNTN(I) = DELGN/(CSBVN2 * N2TEMP(I) * ((PVNTN/POPN2) - 1.0))
471      47 WVN2 = WVN2 + WTVNTN(I)
472      IF(WVO2.LT.0.0) WVO2 = 0.0
473      IF(WVN2.LT.0.0) WVN2 = 0.0
474      C
475      90 CONTINUE
476      C
477      C
478      C      *** COMPUTE QUANTITY FLUIDS LOADED INTO TANKS
479      C
480      TOTWTL(1) = 02TOTU + WTRSID(1) + WVO2
481      TOTWTL(2) = N2TOTU + WTRSID(2) + WVN2
482      C
483      C      *** COMPUTE DIAMETER OF FLUID TANKS - ASSUMED SPHERICAL
484      C
485      DITK(1) = ((1.9098 * (TOTWTL(1)/RHOBEG(1)))**0.33) * 12.0
486      DITK(2) = ((1.9098 * (TOTWTL(2)/RHOBEG(2)))**0.33) * 12.0
487      C
488      C      *** COMPUTE FLUID TANK INSULATION WEIGHT
489      C
490      IT1 = SITYPE(1,1)
491      IT2 = SITYPE(2,1)
492      C
493      TIWT(1,1) = NOP(1,1) * ARETK(1) * RHOI(IT1) * SITHIK(1,1)/12.0
494      TIWT(2,1) = NOP(2,1) * ARETK(2) * RHOI(IT2) * SITHIK(2,1)/12.0
495      C
496      C      *** COMPUTE DIAMETER OF FLUID TANK VACUUM JACKETS
497      C
498      DIVJ(1) = DITK + 1.60
499      DIVJ(2) = DITK + 1.85
500      C
501      C      *** COMPUTE WEIGHT OF FLUID TANK PRESSURE VESSELS
502      C
503      MATL1 = SMTYPE(1,1)
504      CALL FINTAB(NTBID(9),MATL1)
505      FTUX1 = MIPE(1,TKOMXT)
506      C
507      THKMT1 = (1.0 * POPO2 * 2.0 * (DITK(1)/2.0))/FTUX1
508      IF(THKMT1.LT.MINTHK(MATL1)) THKMT1 = MINTHK(MATL1)
509      C
510      MATL2 = SMTYPE(2,1)
511      CALL FINTAB(NTBID(9),MATL2)
512      FTUX2 = MIPE(1,TKNMXT)
513      C
514      THKMT2 = (1.0 * POPN2 * 2.0 * (DITK(2)/2.0))/FTUX2
515      IF(THKMT2.LT.MINTHK(MATL2)) THKMT2 = MINTHK(MATL2)
516      C
517      WTPV(1) = 1.35 * ARETK(1) * RHOL(MATL1) * (THKMT1/12.0)
518      C
519      WTPV(2) = 1.35 * ARETK(2) * RHOL(MATL2) * (THKMT2/12.0)
520      C
521      C      *** COMPUTE WEIGHT OF VACUUM JACKETS FOR FLUID TANKS

```

```

***** ECL55 *****
522 C ASSUME SPHERICAL VACUUM JACKET (HARD SHELL) 6061-T6 ALUMINUM
523 C
524 MATL3 = 3
525 CALL FINTAB (NTBID(9),MATL3)
526 FTUX3 = MIPE(1,TENVR)
527 C
528 THKMT3 = (1.0 * (POP2/2.0) * (DIVJ(1)/2.0))/FTUX3
529 IF (THKMT3.LT.MINTHK(MATL3)) THKMT3 = MINTHK(MATL3)
530 C
531 THKMT4 = (1.0 * (POP2/2.0) * (DIVJ(2)/2.0))/FTUX3
532 IF (THKMT4.LT.MINTHK(MATL3)) THKMT4 = MINTHK(MATL3)
533 C
534 WTVJO = (PI * (DIVJ(1)**2)/144.0) * RHOL(MATL3) * (THKMT3/12.0)
535 C
536 WTVJN = (PI * (DIVJ(2)**2)/144.0) * RHOL(MATL3) * (THKMT4/12.0)
537 C
538 *** COMPUTE TOTAL WEIGHT OF TANK
539 C
540 WTTOT(1) = WTPV(1) + WTVJO + TIWT(1,1)
541 WTTOT(2) = WTPV(2) + WTVJN + TIWT(2,1)
542 C
543 *****
544 C
545 *** OUTPUT THE DATA COMPUTED TO THIS POINT ***
546 C
547 JP = PAGE(0)
548 WRITE (IOT,6116)
549 WRITE (IOT,6001)
550 WRITE (IOT,6009)
551 6009 FORMAT(/T3,'DETERMINE FLUID DENSITIES AT FINAL DUTY CYCLE CONDITio
552 INS AND DETERMINE RESIDUAL FLUID QUANTIES')
553 WRITE (IOT,600)
554 WRITE (IOT,647) TKOHXT, TKNHXT
555 647 FORMAT(T3,'TKOHXT =',T13,F8.2,T25,'TKNHXT =',T37,F8.2)
556 WRITE (IOT,648) RHOEND(1), RHOEND(2)
557 648 FORMAT(T3,'RHOEND-02 =',T15,F8.4,T27,'RHOEND-N2 =',T42,F8.4)
558 WRITE (IOT,649) WTRSID(1), WTRSID(2)
559 649 FORMAT(T3,'WTRSID-02 =',T15,F8.3,T27,'WTRSID-N2 =',T42,F8.3)
560 WRITE (IOT,6010)
561 6010 FORMAT(/T3,'DETERMINE FLUID TANK VOLUME AND SURFACE AREAS')
562 WRITE (IOT,600)
563 WRITE (IOT,650) VOLTK(1), VOLTK(2)
564 650 FORMAT(T3,'VOLTK-02 =',T15,F8.3,T27,'VOLTK-N2 =',T42,F8.3)
565 WRITE (IOT,651) ARETK(1), ARETK(2)
566 651 FORMAT(T3,'AREA-02TK =',T15,F8.3,T27,'AREA-N2TK =',T42,F8.3)
567 WRITE (IOT,6011)
568 6011 FORMAT(/T3,'DETERMINE HEAT LEAK INTO FLUID TANKS AND QUANTITY OF V
569 ENTED FLUIDS IF VENTING IS REQUIRED')
570 WRITE (IOT,600)
571 WRITE (IOT,652) TOCND, TNCND
572 652 FORMAT(T3,'TOCND =',T13,F8.4,T25,'TNCND =',T37,F8.4)
573 WRITE (IOT,653) (QO2LK(N),N=1,KCYCLE)
574 653 FORMAT(T3,'QO2LK =',T13,10(F9.6,3X))
575 WRITE (IOT,654) (QN2LK(N),N=1,KCYCLE)
576 654 FORMAT(T3,'QN2LK =',T13,10(F9.6,3X))
577 WRITE (IOT,655) QLKOTK, QLKNTK
578 655 FORMAT(T3,'QLKOTK =',T13,F9.6,T25,'QLKNTK =',T37,F9.6)
579 WRITE (IOT,656) (QREGCO(N),N=1,KCYCLE)

```

```

***** ECLSS *****
580 656 FORMAT(T3,'QREGDO =',T13,10(F8.3,4X))
581   WRITE (IOT,657) (QREGDN(N),N=1,KCYCLE)
582 657 FORMAT(T3,'QREGDN =',T13,10(F8.3,4X))
583   WRITE (IOT,658) (WTVNTO(N),N=1,KCYCLE)
584 658 FORMAT(T3,'WTVNTO =',T13,10(F8.3,4X))
585   WRITE (IOT,659) (WTVNTN(N),N=1,KCYCLE)
586 659 FORMAT(T3,'WTVNTH =',T13,10(F8.3,4X))
587   WRITE (IOT,660) WVO2,WVN2
588 660 FORMAT(T3,'WVO2 =',T13,F8.3,T25,'WVN2 =',T17,F8.3)
589   WRITE (IOT,6012)
590 6012 FORMAT(/T3,'DETERMINE QUANTITY OF FLUIDS LOADED INTO TANKS')
591   WRITE (IOT,600)
592   WRITE (IOT,661) TOTWT(1),TOTWT(2)
593 661 FORMAT(T3,'O2-LOADED =',T17,F8.3,T30,'N2-LOADED =',T44,F8.3)
594   WRITE (IOT,6013)
595 6013 FORMAT(/T3,'DETERMINE PRESSURE VESSEL, VACUUM JACKET, INSULATION,
596   AND TOTAL FLUID TANK WEIGHTS')
597   WRITE (IOT,600)
598   WRITE (IOT,662) DITK(1),DITK(2)
599 662 FORMAT(T3,'DITK-O2=',T15,F8.2,T30,'DITK-N2=',T42,F8.2)
600   WRITE (IOT,663) TIWT(1,1),TIWT(2,1)
601 663 FORMAT(T3,'TIWT-O2=',T15,F8.3,T30,'TIWT-N2=',T42,F8.3)
602   WRITE (IOT,664) DIVJ(1),DIVJ(2)
603 664 FORMAT(T3,'DIVJ-O2=',T15,F8.2,T30,'DIVJ-N2=',T42,F8.2)
604   WRITE (IOT,665) ROFTU(1),ROFTU(2)
605 665 FORMAT(T3,'RHOFTU-O2=',T15,F8.7,T30,'RHOFTU-N2=',T42,F8.7)
606   WRITE (IOT,666) WTPV(1),WTPV(2)
607 666 FORMAT(T3,'WTPV-O2 =',T15,F8.2,T30,'WTPV-N2 =',T42,F8.2)
608   WRITE (IOT,667) WTVJ0,WTVJN
609 667 FORMAT(T3,'WTVJ-O2 =',T15,F8.2,T30,'WTVJ-N2 =',T42,F8.2)
610   WRITE (IOT,668) WTTOT(1),WTTOT(2)
611 668 FORMAT(T3,'WTTOT-O2 =',T15,F8.2,T30,'WTTOT-N2 =',T42,F8.2)
612 C
613 C *****
614 C
615 C *** COMPUTE WEIGHT OF O2 HEAT EXCHANGER
616 C
617   JX = 0
618   JX = JX + 1
619   IGAS = 1
620   IFIN = 0
621 C
622   WDOTX(JX,IGAS) = WDOTMX
623   ICODE(JX,IGAS) = HXCODE(JX,IGAS)
624   HEXCIT(JX,IGAS) = TEMPO2
625   HEXCOT(JX,IGAS) = TLSNOM(1)
626   HEXCIP(JX,IGAS) = POPO2
627   H5OREQ(JX,IGAS) = GDOTMX
628   ELCPOW(JX,IGAS) = HWTOMX
629 C
630   CALL HEXELC( 1,HEXCIT(JX,IGAS),HEXCOT(JX,IGAS),HEXCIP(JX,IGAS),
631   1 HTRFLX(1),LINDIA(1),WDOTX(JX,IGAS),O2RHO(IMAX),IFIN,
632   2 WTTOT(JX,IGAS),HXCOLP(JX,IGAS),UOA(JX,IGAS),DH(JX,IGAS),
633   3 HLNTH(JX,IGAS))
634 C
635 C
636 C *** COMPUTE WEIGHT OF N2 HEAT EXCHANGER
637 C

```

```

***** ECLSS *****
638      IGAS = 2
639      IFIN = 0
640      C      B NO FINS ON ELECTRIC HEX
641      WDOTX(JX,IGAS) = WDTNMX
642      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
643      HEXCIT(JX,IGAS) = TEMPN2
644      HEXCOT(JX,IGAS) = TLSNOM(2)
645      HEXCIP(JX,IGAS) = POPN2
646      HSGREQ(JX,IGAS) = QDTNMX
647      ELCPOW(JX,IGAS) = HWTNMX
648      C
649      CALL HEXELC( 18,HEXCIT(JX,IGAS),HEXCOT(JX,IGAS),HEXCIP(JX,IGAS),
650      1 HTRFLX(1),LINDIA(2),WDOTX(JX,IGAS),Q2RHO(JMAX),IFIN,
651      2 WHXTOT(JX,IGAS),HXCDLP(JX,IGAS),UOA(JX,IGAS),DH(JX,IGAS),
652      3 NLNGTH(JX,IGAS))
653      C
654      C      *** OUTPUT THE HEAT EXCHANGER DATA ***
655      C
656      CALL OTHXE
657      C
658      C      *****
659      C
660      C      *** OUTPUT THE POWER SUMMARY DATA ***
661      C
662      CALL OPTPOW
663      C
664      C      *****
665      C
666      C      *** COMPUTE THE TANK ENERGY HISTORY AND HEATER DUTY CYCLE ***
667      C
668      TIMINC = 1.0
669      PTANK1 = POP02
670      PTANK2 = POPN2
671      C
672      LI = 0
673      DO 100 KI = 1,NDCCYCL*2
674      LI = LI + 1
675      TIM(LI) = DCYCLE(KI)
676      100 CONTINUE
677      C
678      LB = 0
679      DO 150 JI = 2,NDCCYCL*2
680      LB = LB + 1
681      THONOP(LB) = DCYCLE(JI)
682      150 CONTINUE
683      C
684      TIME = 0.0
685      TKOW = 0.0
686      TKNW = 0.0
687      QICUMH = 0.0
688      Q2CUMN = 0.0
689      C
690      DO 110 K = 1,KCYCLE
691      C
692      IF(TIM(K).EQ.0.0) GO TO 160
693      C
694      NTP = TIM(K) * 6.0
695      WDT030 = WDT02(K)/6.0

```



```

***** ECLSS *****
696      WDTN30 = WDTN2(K)/6.0
697      JP = PAGE(0)
698      C
699      WRITE (IOT,700)
700      700 FORMAT(/T42,'*** COMPUTE TANK ENERGY HISTORY AND HEATER DUTY CYCLE
701      | ***|/)
702      C
703      WRITE (IOT,701) K, TIM(K)
704      701 FORMAT(/T5,'SYSTEM DUTY CYCLE NUMBER ',I4,T45,'CYCLE IS ',F6.1,' H
705      |OURS IN LENGTH|)
706      C
707      GO TO 170
708      C
709      160 CONTINUE
710      IF(TNONOP(K).NE.0.0) GO TO 161
711      JP = PAGE(0)
712      WRITE (IOT,700)
713      WRITE (IOT,162)
714      162 FORMAT(T5,' ***** THERE IS AN ERROR IN THE DUTY CYCLE DATA INPUT *
715      |*****| /T5,' ** THE CALCULATION WILL CONTINUE, OMITTING THE NON-OPE
716      |RATING PERIOD **|)
717      GO TO 110
718      C
719      161 CONTINUE
720      C
721      JP = PAGE(0)
722      WRITE (IOT,700)
723      WRITE (IOT,701) K, TNONOP(K)
724      WRITE (IOT,163) TNONOP(K)
725      163 FORMAT(T5,' *** THE SYSTEM IN A NON-DEMAND DUTY CYCLE PERIOD FOR
726      |',F6.2,' HOURS, /T5,' ** THE ANALYSIS FOR THIS PERIOD CONSIDERS ON
727      |2LY SPACECRAFT GAS LEAKAGE|)
728      C
729      NTP = TNONOP(K) * 6.0
730      WDT030 = (0.21 * GLKRAT/24.0)/6.0
731      WDTN30 = (0.79 * GLKRAT/24.0)/6.0
732      C
733      170 CONTINUE
734      C
735      WRITE (IOT,702) K
736      702 FORMAT(/T3,'TIME',T11,'GAS ',T20,'PER-CENT',T33,'DENSITY',T44,
737      |1 'FLUID',T53,'THETA',T63,'PHI',T71,'Q/REQD',T81,'Q-CUM',T91,'TANK',
738      |2,T100,'Q-HTR.',T110,'HEATER',T11,'FLOW',T20,'WITHDRAWN',T44,'TEMP.',
739      |3,T91,'PRES.',T100,'REQD.',T110,'TIME-ON',T3,'(MIN)',T11,'(LBS)',
740      |4,T20,'(PERCENT)',T33,'(LB/CF)',T45,'(-R-)',T53,'(B/LB)',T62,
741      |5 '(P-CF/B)',T72,'(BTU)',T81,'(BTU)',T91,'(PSIA)',T100,'(BTU)',
742      |6 T110,'(MIN.)',T11,'O2/N2',T22,'O2/N2',
743      |7 T34,'O2/N2',T44,'O2/N2',T53,'O2/N2',T62,'O2/N2',T72,'O2/N2',T81,
744      |8 'O2/N2',T91,'O2/N2',T100,'O2/N2',T110,'O2/N2',T120,'CYCLE -',I4,))
745      C
746      LPRES = 0
747      IK = 0
748      C
749      DO 120 I = 1,NTP
750      C
751      IF(LPRES.EQ.1) GO TO 112
752      IF(RPRTIM(K).GT.0.0) GO TO 111
753      C

```

```

***** ECL55 *****
754      TIME = TIME + 10.0
755      C
756      TKOW = TKOW + WDT030
757      TKODP = TKOW
758      TKNW = TKNW + WDTN30
759      TKNDP = TKNW
760      GO TO 113
761      C
762      111 LPRES = LPRES + 1
763      RPTIME = RPRTIM(K) * 60.0
764      116 CONTINUE
765      TIME = TIME + (RPTIME/10.0)
766      IK = IK + 1
767      TKOW = TKOW + ((WDT02(K) * RPRTIM(K))/10.0)
768      TKODP = TKOW
769      TKNW = TKNW + ((WDTN2(K) * RPRTIM(K))/10.0)
770      TKNDP = TKNW
771      WDT030 = (WDT02(K) * RPRTIM(K))/10.0
772      WDTN30 = (WDTN2(K) * RPRTIM(K))/10.0
773      IF (IK.EQ.10) IK = 0
774      GO TO 113
775      C
776      112 TIME = TIME + 10.0
777      C
778      TKOW = TKOW + WDOTN(K)/6.0
779      TKODP = TKOW
780      TKNW = TKNW + WDOTNN(K)/6.0
781      TKNDP = TKNW
782      WDT030 = WDOTN(K)/6.0
783      WDTN30 = WDOTNN(K)/6.0
784      GO TO 113
785      C
786      C
787      C
788      113 CONTINUE
789      C
790      PCOXW = TKODP/OZTOTU
791      PCN2W = TKNDP/NZTOTU
792      C
793      C
794      C1 = 144.0/(RHOBE(1) * (1544.2546/31.9988))
795      C2 = 144.0/(RHOBE(2) * (1544.2546/28.0134))
796      C3 = 1.0 - ((C1 * TKFPRS(1))/(ZFO * TKFTEM(1)))
797      C4 = 1.0 - ((C2 * TKFPRS(2))/(ZFN * TKFTEM(2)))
798      C
799      C
800      ORHO = RHOBE(1) * (1.0 - (PCOXW * C3))
801      NRHO = RHOBE(2) * (1.0 - (PCN2W * C4))
802      C
803      CALL FINTAB (NTRID(8))
804      XTAB(1) = PTANK1
805      XTAB(2) = ORHO
806      OXTEM = HIPE(2,XTAB)
807      C
808      CALL FINTAB (NTRID(41))
809      XTAB(1) = PTANK2
810      XTAB(2) = NRHO
811      N2TEM = HIPE(2,XTAB)

```

```

***** ECLSS *****
012 C
013 C
014 CALL PHTON(OTEM,ORHO,1,PHI,THETA)
015 DQDM1 = THETA
016 DPDU1 = PHI
017 C
018 C
019 CALL PHTON(N2TEM,NRHO,18,PHI,THETA)
020 DQDM2 = THETA
021 DPDU2 = PHI
022 C
023 C
024 QDTTK1 = WDT030 * DQDM1
025 QDTTK2 = WDTN30 * DQDM2
026 Q1CUM0 = Q1CUM0 + QDTTK1
027 Q2CUMN = Q2CUMN + QDTTK2
028 C
029 CALL BETAB(OTEM,ORHO,1,BETA0)
030 CALL BETAB(N2TEM,NRHO,18,BETAN)
031 C
032 CALL CSUBP(OTEM,PTANK1,1,CPO)
033 CALL CSUBP(N2TEM,PTANK2,18,CPN)
034 C
035 C
036 DELP1 = TIMINC * ((DPDU1/VOLTK(1)) * ((-CPO/BETA0)*WDT030))
037 DELP2 = TIMINC * ((DPDU2/VOLTK(2)) * ((-CPN/BETAN)*WDTN30))
038 C
039 PTANK2 = PTANK2 + DELP2
040 PTANK1 = PTANK1 + DELP1
041 C
042 IF(HTRFLX(2).LE.0.0) GO TO 122
043 C
044 QHTR1 = PI * HTRDIA(1) * HTRLNG(1) * HTRFLX(2)/60.0      # PER MIN.
045 QHTR2 = PI * HTRDIA(2) * HTRLNG(2) * HTRFLX(2)/60.0
046 GO TO 123
047 C
048 122 CONTINUE
049 C
050 QHTR1 = (PI * HTRDIA(1) * HTRLNG(1) * HTRRA1)/60.0      # PER MIN.
051 QHTR2 = (PI * HTRDIA(2) * HTRLNG(2) * HTRRA2)/60.0      # PER MIN.
052 C
053 123 CONTINUE
054 C
055 IF(PTANK1.LE.PSET1) GO TO 130
056 GO TO 135
057 130 QELC1 = Q1CUM0
058 HTRON1 = QELC1/QHTR1
059 135 IF(PTANK2.LE.PSET2) GO TO 140
060 GO TO 118
061 140 QELC2 = Q2CUMN
062 HTRON2 = QELC2/QHTR2
063 C
064 118 CONTINUE
065 C
066 IF(PAGE(2)) GO TO 227
067 GO TO 228
068 227 WRITE (10T,702) K
069 JP = PAGE(6)

```

```

***** ECLSS *****
870      228 CONTINUE
871      C
872      WRITE (107,703) TIME,WDTO30 ,PCOXW,ORHO,OXTEM,DQDM1,DPDU1,GDTTK1,
873      1 QICUM0,PTANK1,GELC1,HTRON1,WDTN30 ,PCN2W,NRHO,N2TEM,DQDM2,
874      2 DPDU2,GDTTK2,Q2CUMN,PTANK2,GELC2,HTRON2
875      703 FORMAT(T1,F9.2,T11,F6.3,T21,F6.5,T32,F8.5,T43,F6.2,T52,F6.2,T61,
876      1 F6.2,T71,F6.1,T80,F7.0,T90,F7.2,T99,F7.0,T109,F7.2,T11,F6.3,T21,
877      2 F6.5,T32,F8.5,T43,F6.2,T52,F6.2,T61,F6.2,T71,F6.1,T80,F7.0,T90,
878      3 F7.2,T99,F7.0,T109,F7.2)
879      C
880      IF (PTANK1,LE,PSET1) HTRON1 = 0.0
881      IF (PTANK1,LE,PSET1) GELC1 = 0.0
882      IF (PTANK1,LE,PSET1) QICUM0 = 0.0
883      IF (PTANK1,LE,PSET1) PTANK1 = POP02
884      IF (PTANK2,LE,PSET2) HTRON2 = 0.0
885      IF (PTANK2,LE,PSET2) GELC2 = 0.0
886      IF (PTANK2,LE,PSET2) Q2CUMN = 0.0
887      IF (PTANK2,LE,PSET2) PTANK2 = POPN2
888      C
889      IF (IK,GT,0) GO TO 116
890      C
891      120 CONTINUE
892      C
893      110 CONTINUE
894      C
895      C *****
896      C
897      C
898      C *****
899      C
900      C *** DO THE ECLSS CONFIGURATION ANALYSIS ***
901      C
902      CALL LSSCHP
903      C
904      C *****
905      C
906      WRITE (107,6099)
907      6099 FORMAT(////T25,12(1*****)/T25,12(1*****)//T25,1***** THE SUP
908      IERCRITICAL EC-LSS CALCULATIONS HAVE BEEN COMPLETED *****//T25,
909      2 12(1*****)/T25,12(1*****)//)
910      C
911      RETURN
912      C
913      END

```

***** SUBROUTINE ENGINE

```

1      C      *****
2      C      * ROUTINE NAME = ENGINE WEIGHT AND TOTAL *
3      C      * IMPULSE PROPELLANT WEIGHT *
4      C      * CALCULATION ROUTINE *
5      C      * ROUTINE LANG = FORTRAN V UNIVAC 1108 EXEC 2 *
6      C      * PROGRAMMER = R. BOLLINGER 1949 102 26999 *
7      C      * DATE CODED = 3/9/70 *
8      C      *****
9      C
10     C      SUBROUTINE ENGINE
11
12     C      LOGICAL DIAG
13     C      LOGICAL PAGE,JP
14
15     C      INCLUDE CCNTRL
16     C      INCLUDE CDCYCL
17     C      INCLUDE CENG
18     C      INCLUDE CIOUNT
19     C      INCLUDE TABLOK
20
21     C      ***** CALCULATE THE ENGINE WEIGHT
22
23     C      IF (DIAG(0,6HENGINE)) WRITE (IOT,6000) NENG,GITEMP,THRUST,PSUBC,
24     C      EXPRAT
25
26     C      IF(SYSNUM,EQ,5) GO TO 5
27
28     C      CALL FINTAB (NTBID(1))
29     C      XTAB(1) = GITEMP
30     C      XTAB(2) = PSUBC
31     C      XTAB(3) = THRUST
32     C      ENGHT = MIPE(3,XTAB) * NENG
33
34     C      ***** CALCULATE THE SPECIFIC IMPULSE
35
36     C      CALL FINTAB (NTBID(2))
37     C      XTAB(1) = GITEMP
38     C      XTAB(2) = MIXRAT
39     C      ISP = MIPE(2,XTAB)
40
41     C      GO TO 15
42
43     C      5 CONTINUE
44
45     C      ***** CALCULATE THE OMS ENGINE WEIGHT
46
47     C      CALL FINTAB (NTBID(10))
48     C      XTAB(1) = PSUBC
49     C      XTAB(2) = THRUST
50     C      ENGHT = MIPE(2,XTAB) * NENG
51
52     C      ***** CALCULATE OMS SPECIFIC IMPULSE
53
54     C      CALL FINTAB (NTBID(11))
55     C      XTAB(1) = PSUBC
56     C      XTAB(2) = MIXRAT
57     C      ISP = MIPE(2,XTAB)

```

***** ENGINE *****

```

58 C
59 15 CONTINUE
60 C
61 C
62 C ***** CALCULATE THE FLOW RATE
63 C ***** CALCULATE THE TOTAL IMPULSE PROPELLANT WEIGHT
64 C
65 KNT = 0
66 TIPWT = 0.0
67 WDOT = 0.0
68 C
69 DO 10 I1 = 1, NDCYCL, 2
70 KNT = KNT + 1
71 IF (DCYCLE(I1)) .10,
72 WDOT1 = IARS(NEOP(KNT))*THRUST/(PS1(KNT)*ISP)
73 TIPWT = TIPWT + WDOT1*DCYCLE(I1)
74 WDOTJ(KNT,2) = WDOT1/(MIXRAT + 1.0)
75 WDOTJ(KNT,1) = WDOT1 - WDOTJ(KNT,2)
76 IF (NEOP(KNT).LT.0) GO TO 10
77 WDOT = AMAX1(WDOT,WDOT1)
78 10 CONTINUE
79 WDOT1(2) = WDOT/(MIXRAT + 1.0)
80 WDOT1(1) = WDOT - WDOT1(2)
81 C
82 IF (DIAG(1,6HENGINE)) WRITE (10T,6000) KNT,ISP,WDOT,TIPWT,ENGWT,
83 WDOT1,((WDOTJ(I,J),I=1,KNT),J=1,2)
84 C
85 JP = PAGE(0)
86 C
87 WRITE (6,6100)
88 WRITE (6,6301) ISP,ENGWT,WDOT1,WDOT1(1),WDOT1(2),TIPWT
89 RETURN
90 6000 FORMAT ('+',14X,15,10X,6E15.6/(8E15.6))
91 6100 FORMAT ('//T38, '*** INITIATE PROGRAM AND CHARACTERIZE CONSUMER PARAM
92 IETERS ***//')
93 6301 FORMAT (T60, 'COMPUTED ENGINE PARAMETERS *//T45, 'ENGINE ISPI,
94 1 T80,E15.8/T45, 'ENGINE WEIGHT = (LRS),1,T80,E15.8/T45, 'TOTAL ENGINE
95 2 FLOW = (LB/SEC)',T80,E15.8/T45, 'ONE ENGINE OXID.FLOW RATE=(LB/SEC
96 3)',T80,E15.8/T45, 'ONE ENGINE FUEL FLOW RATE=(LB/SEC)',T80,E15.8/T4
97 45, 'THRUST IMPULSE PROPELLANT WGT.',T80,E15.8)
98 C
99 END

```

***** FUNCTION FINDR

```
1      FUNCTION FINDR(N)
2      DIMENSION G(18)
3      DATA G /48.31,766.8,40.67,766.8,42.01,96.35,28.62,
4      1 96.35,28.62,55.81,40.67,90.77,16.78,37.0,11.90,33.50,386.3,
5      2 55.161/
6      FINDR=G(N)
7      RETURN
8      END
```

***** FUNCTION FING1

```

1      FUNCTION FING1(T,D)
2      COMMON/CEOS/G(41)
3      COMMON /SCRH/ X(40)
4      C
5      C.... ROUTINE TO CALCULATE INTEGRAL((R/D-1/D**2(DP/DT)) DD)
6      C
7      C.... WRITTEN 7/18/71 A MYERS
8      C
9      D2=D*D
10     D3=D2*D
11     D4=D3*D
12     D5=D4*D
13     D6=D5*D
14     D7=D6*D
15     D8=D7*D
16     D9=D8*D
17     D10=D9*D
18     TS= SQRT(T)
19     T2=T*T
20     T3=T2*T
21     T4=T3*T
22     T5=T4*T
23     GM=G(41)
24     F= EXP(GM*D2)
25     G1=F/(2.00*GM)
26     G2=(F*D2-2.00*G1)/(2.00*GM)
27     G3=(F*D4-4.00*G2)/(2.00*GM)
28     G4=(F*D6-6.00*G3)/(2.00*GM)
29     G5=(F*D8-8.00*G4)/(2.00*GM)
30     G6=(F*D10-10.00*G5)/(2.00*GM)
31     X( 1)=-D
32     X( 2)=-D/(2.00*TS)
33     X( 3)=0.0
34     X( 4)=D/T2
35     X( 5)=2.00*D/T3
36     X( 6)=-D2/2.00
37     X( 7)=0.0
38     X( 8)=D2/(2.00*T2)
39     X( 9)=D2/T3
40     X(10)=-D3/3.00
41     X(11)=0.0
42     X(12)=D3/(3.00*T2)
43     X(13)=0.0
44     X(14)=D5/(5.00*T2)
45     X(15)= 2.00*D5/(5.00*T3)
46     X(16)=D6/(6.00*T2)
47     X(17)=D7/(7.00*T2)
48     X(18)=2.00*D7/(7.00*T3)
49     X(19)=D8/(4.00*T3)
50     X(20)=2.00*G1/T3
51     X(21)=3.00*G1/T4
52     X(22)=2.00*G2/T3
53     X(23)=4.00*G2/T5
54     X(24)=2.00*G3/T3
55     X(25)=3.00*G3/T4
56     X(26)=2.00*G4/T3
57     X(27)=4.00*G4/T5

```



```

***** FINGI *****
58      X(28)=2.00*G5/T3
59      X(29)=3.00*G5/T4
60      X(30)=2.00*G6/T3
61      X(31)=3.00*G6/T4
62      X(32)=4.00*G6/T5
63      FINGI=0.00
64      DO I I=1,32
65      I FINGI=FINGI+G(I)*X(I)
66      RETURN
67      END

```

***** FUNCTION FING2

```

1      FUNCTION FING2(T,D)
2      COMMON/CEOS/G(41)
3      COMMON /SCRH/ X(40)
4      C
5      C.... ROUTINE TO CALCULATE INTEGRAL((P/D**2-RT/D) DD)
6      C
7      D2=D*D
8      D3=D2*D
9      D4=D3*D
10     D5=D4*D
11     D6=D5*D
12     D7=D6*D
13     D8=D7*D
14     D9=D8*D
15     D10=D9*D
16     TS= SQRT(T)
17     T2=T*T
18     T3=T2*T
19     T4=T3*T
20     T5=T4*T
21     GM=G(41)
22     F= EXP(GM*D2)
23     G1=F/(2.00*GM)
24     G2=(F*D2-2.00*G1)/(2.00*GM)
25     G3=(F*D4-4.00*G2)/(2.00*GM)
26     G4=(F*D6-6.00*G3)/(2.00*GM)
27     G5=(F*D8-8.00*G4)/(2.00*GM)
28     G6=(F*D10-10.00*G5)/(2.00*GM)
29     X( 1)=D*T
30     X( 2)=D*TS
31     X( 3)=D
32     X( 4)=D/T
33     X( 5)=D/T2
34     X( 6)=D2*T/2.00
35     X( 7)=D2/2.00
36     X( 8)=D2/(2.00*T)
37     X( 9)=D2/(2.00*T2)
38     X(10)=D3*T/3.00
39     X(11)=D3/3.00
40     X(12)=D3/(3.00*T)
41     X(13)=D4/4.00
42     X(14)=D5/(5.00*T)
43     X(15)=D5/(5.00*T2)
44     X(16)=D6/(6.00*T)
45     X(17)=D7/(7.00*T)
46     X(18)=D7/(7.00*T2)
47     X(19)=D8/(8.00*T2)
48     X(20)=G1/T2
49     X(21)=G1/T3
50     X(22)=G2/T2
51     X(23)=G2/T4
52     X(24)=G3/T2
53     X(25)=G3/T3
54     X(26)=G4/T2
55     X(27)=G4/T4
56     X(28)=G5/T2
57     X(29)=G5/T3

```

```

*****  FING2  *****
58      X(30)=G6/T2
59      X(31)=G6/T3
60      X(32)=G6/T4
61      FING2=0.00
62      DO I I=1,32
63      I FING2=FING2+G(I)*X(I)
64      RETURN
65      END

```

FUNCTION FING3

```

1      FUNCTION FING3(T,D)
2      COMMON/CE05/G(41)
3      COMMON /SCRH/ X(40)
4      C
5      C.... ROUTINE TO CALCULATE INTEGRAL((T/D**2)*(D2P/DT2) DD)
6      C
7      D2=D*D
8      D3=D2*D
9      D4=D3*D
10     D5=D4*D
11     D6=D5*D
12     D7=D6*D
13     D8=D7*D
14     D9=D8*D
15     D10=D9*D
16     T5= SQRT(T)
17     T2=T*T
18     T3=T2*T
19     T4=T3*T
20     T5=T4*T
21     GM=G(41)
22     F= EXP(GM*D2)
23     G1=F/(2.00*GM)
24     G2=(F*D2-2.00*G1)/(2.00*GM)
25     G3=(F*D4-4.00*G2)/(2.00*GM)
26     G4=(F*D6-6.00*G3)/(2.00*GM)
27     G5=(F*D8-8.00*G4)/(2.00*GM)
28     G6=(F*D10-10.00*G5)/(2.00*GM)
29     X( 1)=0.0
30     X( 2)=-D/(4.00*T5)
31     X( 3)=0.0
32     X( 4)=2.00*D/T2
33     X( 5)=6.00*D/T3
34     X( 6)=0.000
35     X( 7)=0.0
36     X( 8)=D2/T2
37     X( 9)=3.00*D2/T3
38     X(10)=0.000
39     X(11)=0.0
40     X(12)=(2.00*D3)/(3.00*T2)
41     X(13)=0.0
42     X(14)=(2.00*D5)/(5.00*T2)
43     X(15)=(6.00*D5)/(5.00*T3)
44     X(16)=D6/(3.00*T2)
45     X(17)=(2.00*D7)/(7.00*T2)
46     X(18)=(6.00*D7)/(7.00*T3)
47     X(19)=(3.00*D8)/(4.00*T3)
48     X(20)=6.000*G1/T3
49     X(21)=12.00*G1/T4
50     X(22)=6.000*G2/T3
51     X(23)=20.00*G2/T5
52     X(24)=6.000*G3/T3
53     X(25)=12.00*G3/T4
54     X(26)=6.000*G4/T3
55     X(27)=20.00*G4/T5
56     X(28)=6.000*G5/T3
57     X(29)=12.00*G5/T4

```

```

***** FING3 *****
58      X(30)=6.000*G6/T3
59      X(31)=12.00*G6/T4
60      X(32)=20.00*G6/T5
61      FING3=0.00
62      DO I I=1,32
63      I FING3=FING3+G(I)*X(I)
64      RETURN
65      END

```

SUBROUTINE FINTAB

```

1  C      * * * * *
2  C      * ROUTINE NAME - FIND AND INPUT THE MASTER *
3  C      *      TABLE OF A SET OF TABLES *
4  C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
5  C      * PROGRAMMER - R. BOLLINGER 1949 102 26933 *
6  C      * DATE CODED - 3/9/70 *
7  C      * * * * *
8  C
9  C      SUBROUTINE FINTAB(MTN)
10 C
11 C      ***** EXPLANATION OF THE CALLING SEQUENCE
12 C      *
13 C      * MTN = MASTER TABLE NUMBER, THE INDEX OF THE ARRAY TLA
14 C      *      (SEE PDP CTAB) WHICH POINTS TO THE PROPER DRUM
15 C      *      LOCATION FOR THE TABLE.
16 C      *****
17 C
18 C      LOGICAL DIAG
19 C
20 C      INCLUDE CIOUNT
21 C      INCLUDE CTAB
22 C      INCLUDE CTABA
23 C
24 C      IF (DIAG(0,6HFINTAB)) WRITE (10T,6010) MTN,TLA(MTN)
25 C      JTABID = MTN
26 C      IF (TLA(MTN) .GT. 0) GO TO 5
27 C      WRITE (10T,6020) MTN
28 C      CALL EXIT
29 C      5 CONTINUE
30 C      IDX1 = TLA(MTN)
31 C      ND = ITABLE(IDX1)
32 C      IF (ND,LE,2) GO TO 20
33 C
34 C      NDM2 = ND - 2
35 C      DO 10 I1=1,NDM2
36 C      IDX1 = IDX1 + 1
37 C      NP = ITABLE(IDX1)
38 C      ITAB(1,I1) = NP
39 C      DO 10 I2=1,NP
40 C      IDX1 = IDX1 + 1
41 C      TAB(I2+1,I1) = TABLE(IDX1)
42 C      10 CONTINUE
43 C
44 C      20 IDX1 = IDX1 + 1
45 C      IF (DIAG(1,6HFINTAB)) WRITE (10T,6010) ND,((TAB(I,J),J=1,5),I=1,6)
46 C      RETURN
47 C
48 C      6010 FORMAT ('+114X,17,5115/(30X5E15.6)')
49 C
50 C      6020 FORMAT ('0'// 20X '*** LOAD TABLE NO. 114, BEFORE YOU TRY TO USE'
51 C      ' IT.. ***')
52 C      END

```

B-147

LMSC-A991396

FUNCTION FLODEQ

```

1      C      * * * * *
2      C      * ROUTINE NAME - FAND LINE - ONE DIMENSIONAL *
3      C      * COMPRESSIBLE FLOW FUNCTION *
4      C      * FOR ADIABATIC FLOW AT *
5      C      * CONSTANT AREA WITH FRICTION *
6      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
7      C      * PROGRAMMER - R. BOLLINGER 1943 102 26933 *
8      C      * DATE CODED - 3/10/70 *
9      C      * * * * *
10     C
11     C      FUNCTION FLODEQ(M,G)
12     C
13     C      ***** EXPLANATION OF THE CALLING SEQUENCE
14     C      *
15     C      * H - MACH NUMBER
16     C      *
17     C      * G - SPECIFIC HEAT AT CONSTANT PRESSURE/
18     C      * SPECIFIC HEAT AT CONSTANT VOLUME
19     C      *****
20     C      REAL M,MSQ
21     C
22     C      MSQ = M**2
23     C      FLODEQ = (1.0 - MSQ)/(G*MSQ) + (G + 1.0)*ALOG((G + 1.0)*MSQ/
24     C      (2.0 + (G - 1.0) * MSQ))/(2.0 * G)
25     C      RETURN
26     C      END

```

SUBROUTINE FLORAT

```

1      C      * * * * *
2      C      * ROUTINE NAME - TOTAL FLOW RATE FOR H2 AND *
3      C      *      O2 COMPUTATION. *
4      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
5      C      * PROGRAMMER - J.A. MCKAY 19-43 201 X45178 *
6      C      * DATE CODED - FEBRUARY 1973 (REVISED) *
7      C      * * * * *
8      C
9      SUBROUTINE FLORAT
10     C
11     LOGICAL DIAG
12     REAL ICIN,ICOUT
13     REAL K1,K2,K3,K4,K5,K6,K7,K8
14     C
15     INCLUDE CCTRL
16     INCLUDE CENG
17     INCLUDE CFLPAT
18     INCLUDE CHEX
19     INCLUDE CPUMP
20     INCLUDE CTANK
21     INCLUDE CTURBN
22     INCLUDE CIOUNT
23     C
24     DIMENSION EQ13(2),EQ24(2),EQ57(2),EQ68(2)
25     C
26     EQUIVALENCE (EQ13,K1),(EQ24,K2),(EQ13(2),K3),(EQ24(2),K4)
27     , (EQ57,K5),(EQ68,K6),(EQ57(2),K7),(EQ68(2),K8)
28     C
29     C
30     IF (DIAG(0,6HFLORAT)) WRITE (IOT,6020) WDOT1, TOTEMP, TITEMP, TEFF,
31     EPDELP, THRATO
32     IF (DIAG(0,6HFLORAT)) WRITE (IOT,6020) HEXHOT(1,1),HEXHOT(1,2),
33     1 HEXCOT(1,1), HEXCOT(1,2), HXMRAT(1,1), HXMRAT(1,2), HEXHIP(1,1)
34     2, HEXHIP(1,2)
35     C
36     C
37     I3 = 1
38     ISFLG = 1
39     IF (SYSNUM .EQ. 2) I3 = 2
40     IF (SYSNUM .EQ. 2 .AND. SCRIT .EQ. 2) ISFLG = 2
41     DO 20 I1=1,2
42     GO TO (14,12),ISFLG
43     12 TERM1 = 0.
44     TERM2 = 0.
45     IF (HEXHIP(2,I1) .EQ. 0.) GO TO 18
46     14 CONTINUE
47     C      TERM FOR HEAT EXCHANGER NO. 1
48     AVGTH = (HEXHIT(I3,I1) + HEXHOT(I3,I1)) / 2.0
49     CALL CSUBP1 (AVGTH,HXMRAT(I3,I1),CPHEX)
50     DIH = CPHEX * (HEXHIT(I3,I1) - HEXHOT(I3,I1))
51     CALL ENTHOH (HEXCIP(I3,I1),HEXCIT(I3,I1),I1,ICIN)
52     CALL ENTHOH (HEXCOP(I3,I1),HEXCOT(I3,I1),I1,ICOUT)
53     DIC = ICOUT - ICIN
54     TERM1 = DIC / (DIH*(1.0+HXMRAT(I3,I1)))
55     GO TO (16,18),ISFLG
56     16 CONTINUE
57     C      TERM FOR PUMP TURBINE

```


***** FLORAT *****

```

58      AVGTT = (TTEMP(11) + TOTEMP(11)) / 2.0
59      CALL CSUBPI (AVGTT,THRATO(11),CPEP)
60      DELMTP = CPEP * (TTEMP(11) - TOTEMP(11))
61      CALL RHOLIO (SITEMP(11,1),1,RHOLG)
62      TERM2 = 0.185*EPDELP(11) / (RHOLG*PEFF(11)*TEFF(11)*DELMTP
63      |
64      | 18 CONTINUE
65      C
66      C      CALCULATE THE K-TERMS
67      EQ13(11) = TERM1 * HXMRAT(13,11)
68      EQ24(11) = TERM1
69      EQ57(11) = TERM2 * THRATO(11)
70      EQ68(11) = TERM2
71      20 CONTINUE
72      C
73      DENMHX = K1 + K4 + K2*K3 - K1*K4 - 1.0
74      DENMTB = K5 + K8 + K6*K7 - K5*K8 - 1.0
75      C
76      WFL03 = (WDOT1 * K4 - WDOT1 * WDOT1(2)*K3) / DENMHX
77      WFL04 = (WDOT1(2)*K1 - WDOT1(2) - WDOT1 * K2) / DENMHX
78      C
79      WFL07 = (WDOT1 * K8 - WDOT1 * WDOT1(2)*K7) / DENMTB
80      WFL08 = (WDOT1(2)*K5 - WDOT1(2) - WDOT1 * K6) / DENMTB
81      C
82      WFL01 = K1 * WFL03
83      WFL02 = K3 * WFL04
84      WFL03 = K2 * WFL03
85      WFL04 = K4 * WFL04
86      C
87      WFL05 = K5 * WFL07
88      WFL06 = K7 * WFL08
89      WFL07 = K6 * WFL07
90      WFL08 = K8 * WFL08
91      C
92      C      SET UP CONDITIONING FLOW RATE EQUATIONS
93      WDHXT0 = WFL01 + WFL02
94      WDHXTF = WFL03 + WFL04
95      C
96      WDTPT0 = WFL05 + WFL06
97      WDTPTF = WFL07 + WFL08
98      C
99      WDOTT(1) = WDOT1(1) + WDHXT0 + WDTPT0
100     WDOTT(2) = WDOT1(2) + WDHXTF + WDTPTF
101     C
102     GO TO (30,32),ISFLG
103     C      OUTPUT TURBINE G.G. FLOWRATES
104     30 CALL OTPELT
105     GO TO 34
106     C      OUTPUT HEAT EXCHANGER G.G. FLOWRATES
107     32 CALL OTPELX
108     C
109     34 CONTINUE
110     C
111     IF (DIAG(1,6HFLORAT)) WRITE (107,6020) K1,K2,K3,K4,K5,K6,K7,K8,
112     |
113     | DENMHX,DENMTB
114     C
115     RETURN
116     C

```

***** FLORAT *****

116 .6020 FORMAT (14/14X,6E15.6/(15X6E15.6))
117 C
118 END

SUBROUTINE FUELCL

```

1      SUBROUTINE FUELCL
2      C
3      LOGICAL JP,PAGE
4      C
5      INCLUDE CACCUH
6      INCLUDE CAPU
7      INCLUDE CDCYCL
8      INCLUDE CENG
9      INCLUDE CFUEL
10     INCLUDE CHEX
11     INCLUDE CHSORC
12     INCLUDE CIOUNT
13     INCLUDE CMATRL
14     INCLUDE CONST
15     INCLUDE CPUMP
16     INCLUDE CTANK
17     INCLUDE TABLOK
18     C
19     DIMENSION TIM(12)
20     DIMENSION WDOTX(MHX,2)
21     EQUIVALENCE (WDOTCF,WDOTX)
22     C
23     *****
24     C
25     JP = PAGE(0)
26     C
27     WRITE (10T,6000)
28     6000 FORMAT(/T38, '*** INITIATE PROGRAM AND CHARACTERIZE FUEL CELL PARA
29     METERS ***')
30     WRITE (10T,6001)
31     6001 FORMAT(//T21 'PARAMETER',T14,'CYCLE-1',T26,'CYCLE-2',
32     1 T38,'CYCLE-3',T50,'CYCLE-4',T62,'CYCLE-5',T74,'CYCLE-6',T86,
33     2 'CYCLE-7',T98,'CYCLE-8',T110,'CYCLE-9',T121,'CYCLE-10'/)
34     C
35     C
36     C *****
37     C
38     BEGIN COMPUTATIONS FOR SUPERCRITICAL STORAGE
39     C
40     C *****
41     C
42     *** SET SELECTED INPUT PARAMETERS FOR CONVENIENCE
43     C
44     PCOFC = SOPRES(1,1)
45     PCHFC = SOPRES(2,1)
46     C
47     *** COMPUTE TOTAL ELECTRICAL POWER SUPPLIED FOR MISSION.
48     C
49     POWTOT = 0.0
50     I = 0
51     DO 5 I = 1,NDCYCL,2
52     I = I + 1
53     POWTOT = POWTOT + PKW(I) * DCYCLE(I) * NEOP(I)  # POWER IN KW-HRS
54     5 CONTINUE
55     C
56     C *** COMPUTE QUANTITY OF REACTANTS CONSUMED FOR POWER
57     C *** COMPUTE FLOWRATES OF EACH REACTANT FOR EACH INTERVAL

```

```

***** FUELCL *****
58 C
59 WRFORP = 0.0
60 I = 0
61 DO 10 I = 1, NDCYCL, 2
62 I = I + 1
63 WRP(I) = PKW(I) * DCYCLE(I) * SRCFC * NEOP(I)
64 WRFORP = WRFORP + WRP(I)
65 C
66 10 CONTINUE
67 C
68 *** OXYGEN CONSUMPTION
69 C
70 C1 = NRFC / (NRFC + 1.0)
71 WCONSO = WRFORP * C1
72 C
73 *** HYDROGEN CONSUMPTION
74 C
75 C2 = 1.0 / (NRFC + 1.0)
76 WHCONSO = WRFORP * C2
77 C
78 DO 11 I = 1, KCYCLE
79 WORFP(I) = WRP(I) * C1
80 WHRFP(I) = WRP(I) * C2
81 C
82 WDTFCO(I) = PKW(I) * SRCFC * C1
83 WDTFCH(I) = PKW(I) * SRCFC * C2
84 11 CONTINUE
85 C
86 *** MAX O2 AND H2 FLOW RATES
87 C
88 PKHMAX = 0.0
89 DO 15 I = 1, KCYCLE
90 IF (PKW(I) .LT. PKHMAX) GO TO 15
91 PKHMAX = PKW(I)
92 15 CONTINUE
93 C
94 WDOTMX(1) = SRCFC * PKHMAX * C1
95 WDOTMX(2) = SRCFC * PKHMAX * C2
96 C
97 WDOTI(1) = WDOTMX(1) / 3600.0
98 WDOTI(2) = WDOTMX(2) / 3600.0
99 C
100 WRITE(IOT, 6004)
101 6004 FORMAT(T4, 'COMPUTE TOTAL POWER AND FLOW RATES')
102 WRITE(IOT, 672)
103 672 FORMAT(' ')
104 C
105 WRITE(IOT, 601) POWTOT, WRFORP
106 601 FORMAT(T4, 'POWTOT=', T13, F8.2, T25, 'WRFORP=', T37, F8.1)
107 WRITE(IOT, 602) (WRP(J), J=1, KCYCLE)
108 602 FORMAT(T4, 'WRP =', T13, 10(F8.3, 4X))
109 WRITE(IOT, 603) C1, C2
110 603 FORMAT(T4, 'C1 =', T13, F8.3, T25, 'C2 =', T37, F8.3)
111 WRITE(IOT, 604) WCONSO, WHCONSO
112 604 FORMAT(T4, 'WCONSO=', T13, F8.2, T25, 'WHCONSO=', T37, F8.2)
113 WRITE(IOT, 674) (WORFP(J), J=1, KCYCLE)
114 674 FORMAT(T4, 'WORFP =', T13, 10(F8.2, 4X))
115 WRITE(IOT, 675) (WHRFP(J), J=1, KCYCLE)

```

```

***** FUELCL *****
116 675 FORMAT(T4,'WHREP =',T13,10(F8.2,4X))
117 WRITE (IOT,605) (WDTFCO(J),J=1,KCYCLE)
118 605 FORMAT(T4,'WDTFCO=',T13,10(F8.2,4X))
119 WRITE (IOT,606) (WDTFCH(J),J=1,KCYCLE)
120 606 FORMAT(T4,'WDTFCH=',T13,10(F8.2,4X))
121 WRITE (IOT,609) WDOTMX(1), WDOTMX(2)
122 609 FORMAT(T4,'WDOTMX-O2 =',F10.3,T30,'WDOTMX-H2 =',F10.3)
123 WRITE (IOT,673) WDOTI(1), WDOTI(2)
124 673 FORMAT(T4,'WDOTI-O2 =',F12.8,T30,'WDOTI-H2 =',F12.8)
125 C
126 C *****
127 C
128 C *** DETERMINE AVERAGE SPECIFIC HEAT OF FREON COOLANT
129 C
130 TMF21 = (TF21IN + TF21OU)/2.0
131 CF21 = CSPF21(TM21)
132 QF21 = CF21 * (TF21IN - TF21OU) B HEAT VALUE BTU/LB
133 C
134 C *****
135 C
136 C *** COMPUTE TOTAL HEAT REJECTED AND HEAT REJECTED FOR EACH
137 C *** OPERATING INTERVAL BY FUEL CELLS
138 C
139 QFCTOT = 0.0
140 I = 0
141 DO 40 I2 = 1,NDCYCL,2
142 I = I + 1
143 QAVAIL(I) = GDTFC * PKW(I) * DCYCLE(I2) * NEOP(I)
144 C
145 WDTF21(I) = QAVAIL(I)/QF21
146 C
147 QFCTOT = QFCTOT + QAVAIL(I)
148 40 CONTINUE
149 C
150 WRITE (IOT,672)
151 WRITE (IOT,6005)
152 6005 FORMAT(T4,'COMPUTE FUEL CELL HEAT REJECTION DATA')
153 WRITE (IOT,672)
154 C
155 WRITE (IOT,607) PKWMAX, TMF21
156 607 FORMAT(T4,'PKWMAX=',T13,F8.2,T25,'TMF21 =',T37,F8.2)
157 WRITE (IOT,608) QF21, CF21
158 608 FORMAT(T4,'QF21 =',T13,F8.2,T25,'CF21 =',T37,F8.2)
159 WRITE (IOT,610) (QAVAIL(J),J=1,KCYCLE)
160 610 FORMAT(T4,'QAVAIL=',T13,10(F10.0,2X))
161 WRITE (IOT,611) (WDTF21(J),J=1,KCYCLE)
162 611 FORMAT(T4,'WDTF21=',T13,10(F10.1,2X))
163 WRITE (IOT,612) QFCTOT
164 612 FORMAT(T4,'QFCTOT=',T13,F10.1)
165 C
166 C *****
167 C
168 C *** DETERMINE ENTHALPY OF FLUIDS FEEDING FUEL CELL
169 C
170 HFCE = OXENTH(PCOFC,TFCNOM(1))
171 HFCH = HYENTH(PCHFC,TFCNOM(2))
172 C
173 C *****

```

```

***** FUELCL *****
174 C
175 C      DETERMINE INITIAL TANK TEMPERATURES
176 C
177 C      CALL FINTAB(NTBID(8))
178 C      XTAB(1) = PCOFC
179 C      XTAB(2) = 70.126
180 C      TEMPO2 = MIPE(2,XTAB)
181 C
182 C      CALL FINTAB(NTBID(7))
183 C      XTAB(1) = PCHFC
184 C      XTAB(2) = 4.365
185 C      TEMPH2 = MIPE(2,XTAB)
186 C
187 C      DETERMINE INITIAL CSUBV VALUES FOR TANK T AND P CONDITIONS
188 C
189 C      CISBVO = CSUBV(TEMPO2,PCOFC,1)
190 C      CISBVII = CSUBV(TEMPH2,PCHFC,2)
191 C
192 C      COMPUTE THE COMPRESSIBILITY OF H2 AT TEMPERATURE TFH AND PRESSURE PFH
193 C
194 C      COMPUTE THE COMPRESSIBILITY OF O2 AT TEMPERATURE TFO AND PRESSURE PFO
195 C
196 C      TKTO = 0.0
197 C      TKTH = 0.0
198 C      I = 0
199 C      DO 450 I1 = 1,NDCYCL*2
200 C      I = I + 1
201 C
202 C      ZFO = ZGET(TFOFC,PFOFC,1)
203 C
204 C      CALL ZFIND(TFHFC,PFHFC,2,ZFH)
205 C
206 C      COMPUTE THE PERCENT OF USABLE H2 AND O2 WITHDRAWN UP TO THIS POINT
207 C      IN THE MISSION
208 C
209 C      TKTO = TKTO + DCYCLE(I1) * WDTFCO(I)
210 C      TKO2WD(I) = TKTO
211 C      TKTH = TKTH + DCYCLE(I1) * WDTFCH(I)
212 C      TKH2WD(I) = TKTH
213 C
214 C      ESTIMATE RESERVES AT 20 PERCENT
215 C
216 C      PCWD02(I) = TKO2WD(I)/(WOCONS + 0.2 * WOCONS)
217 C      PCWDH2(I) = TKH2WD(I)/(WHCONS + 0.2 * WHCONS)
218 C
219 C      COMPUTE THE DENSITY OF H2 AND O2 AS A FUNCTION OF PERCENT WITHDRAWN.
220 C
221 C      C3 = 1.0 - ((0.0427*PFHFC)/(ZFH*TFHFC))
222 C      C4 = 1.0 - ((0.04253*PFOFC)/(ZFO*TFOFC))
223 C
224 C      RHO2(I) = 70.126 * (1.0 - (PCWD02(I) * C4))
225 C      RHOH2(I) = 4.365 * (1.0 - (PCWDH2(I) * C3))
226 C
227 C      *****
228 C
229 C
230 C      COMPUTE THE TEMPERATURE OF O2 IN STORAGE TANK AS A FUNCTION OF
231 C      DENSITY AND PRESSURE.

```

```

***** FUELCL *****
232 C
233 CALL FINTAB (NTBID(8))
234 XTAB(1) = PCOFC
235 XTAB(2) = RHOT02(1)
236 TKO(1) = MIPE(2,XTAB)
237 C
238 C COMPUTE THE TEMPERATURE OF H2 IN STORAGE TANK DURING TIME INTERVAL
239 C THETA(1) AS A FUNCTION OF DENSITY AND CONDITIONED PRESSURE.
240 C
241 CALL FINTAB (NTBID(7))
242 XTAB(1) = PCHFC
243 XTAB(2) = RHOTH2(1)
244 TKH(1) = MIPE(2,XTAB)
245 C
246 C *****
247 C
248 C *** COMPUTE SPECIFIC HEAT INPUT (DQ/DH) FOR O2 AND H2 AS A
249 C * FUNCTION OF DENSITY AND STORED FLUID PRESSURE
250 C
251 CALL PHTHON(TKO(1),RHOT02(1),1,PHI,THETA)
252 DGDWO(1) = THETA
253 C
254 CALL FINTAB (NTBID(5))
255 XTAB(1) = PCHFC
256 XTAB(2) = RHOTH2(1)
257 DGDWH(1) = MIPE(2,XTAB)
258 C
259 C *****
260 C
261 C *** SIZE O2 CONDITIONING HEAT EXCHANGER
262 C
263 HTKO(1) = OXENTH(PCOFC,TKO(1))
264 Q1ODTR(1) = WDTFCO(1) * (HFCO - HTKO(1))
265 WDTIFO(1) = Q1ODTR(1)/QF21
266 C
267 C *** SIZE H2 CONDITIONING HEAT EXCHANGER
268 HTKH(1) = HYENTH(PCHFC,TKH(1))
269 Q1HDTR(1) = WDTFCH(1) * (HFCH - HTKH(1))
270 WDTIFH(1) = Q1HDTR(1)/QF21
271 C
272 C *****
273 C
274 C *** SIZE O2 TANK AND H2 TANK HEAT REQUIREMENTS
275 C
276 C
277 Q2ODTR(1) = WDTFCO(1) * DGDWO(1)
278 C
279 Q2HDTR(1) = WDTFCH(1) * DGDWH(1)
280 C
281 WDT2FO(1) = Q2ODTR(1)/QF21
282 C
283 WDT2FH(1) = Q2HDTR(1)/QF21
284 C
285 C *****
286 C
287 C *** COMPUTE THE ENERGY DIRIVATIVE (PHI) FOR O2 AND H2 AS A
288 C * FUNCTION OF PRESSURE AND TEMPERATURE FOR CONST VOL
289 C

```

```

***** FUELCL *****
290      CSBVFO(I) = CSUBV(TKO(I),PCOFC,1)
291      CSBVFH(I) = CSUBV(TKH(I),PCHFC,2)
292      C
293      CALL PHTON(TKO(I),RHOTO2(I),1,PHI,THETA)
294      PHIF02(I) = PHI
295      C
296      CALL FINTAB (NTBID(40))
297      XTAB(1) = PCHFC
298      XTAB(2) = RHOTH2(I)
299      PHIFH2(I) = HIPE(2,XTAB)
300      C
301      450 CONTINUE
302      WRITE (IOT,672)
303      WRITE(IOT,6006)
304      6006 FORMAT(T4,'DETERMINE TANK CONDITIONS FOR DUTY CYCLE')
305      WRITE (IOT,672)
306      C
307      WRITE (IOT,613) HFCO, HFCH
308      613 FORMAT(T4,'HFCO =',T13,F8.2,T25,'HFCH =',T37,F8.2)
309      WRITE (IOT,614) TEMPO2, TEMPH2
310      614 FORMAT(T4,'TEMPO2=',T13,F8.2,T25,'TEMPH2=',T37,F8.2)
311      WRITE (IOT,615) CISBVO, CISBVH
312      615 FORMAT(T4,'CISBVO=',T13,F8.3,T25,'CISBVH=',T37,F8.3)
313      WRITE (IOT,616) (TKO2WD(J),J=1,KCYCLE)
314      616 FORMAT(T4,'TKO2WD=',T13,10(F8.2,4X))
315      WRITE (IOT,617) (TKH2WD(J),J=1,KCYCLE)
316      617 FORMAT(T4,'TKH2WD=',T13,10(F8.2,4X))
317      WRITE (IOT,618) (PCWD02(J),J=1,KCYCLE)
318      618 FORMAT(T4,'PCWD02=',T13,10(F8.3,4X))
319      WRITE (IOT,619) (PCWDH2(J),J=1,KCYCLE)
320      619 FORMAT(T4,'PCWDH2=',T13,10(F8.3,4X))
321      WRITE (IOT,620) C3, C4
322      620 FORMAT(T4,'C3 =',T13,F8.2,T25,'C4 =',T37,F8.2)
323      WRITE (IOT,621) (RHOTO2(J),J=1,KCYCLE)
324      621 FORMAT(T4,'RHOTO2=',T13,10(F8.3,4X))
325      WRITE (IOT,622) (RHOTH2(J),J=1,KCYCLE)
326      622 FORMAT(T4,'RHOTH2=',T13,10(F8.3,4X))
327      WRITE (IOT,623) (DQDHO(J),J=1,KCYCLE)
328      623 FORMAT(T4,'DQDHO =',T13,10(F8.2,4X))
329      WRITE (IOT,624) (DQDHH(J),J=1,KCYCLE)
330      624 FORMAT(T4,'DQDHH =',T13,10(F8.2,4X))
331      WRITE (IOT,625) (TKO(J),J=1,KCYCLE)
332      625 FORMAT(T4,'TKO =',T13,10(F8.2,4X))
333      WRITE (IOT,626) (TKH(J),J=1,KCYCLE)
334      626 FORMAT(T4,'TKH =',T13,10(F8.2,4X))
335      C
336      JP = PAGE(0)
337      WRITE (IOT,6116)
338      6116 FORMAT(/T42,'*** CONTINUE COMPUTATION OF FUEL CELL PARAMETERS ***'
339      /)
340      WRITE (IOT,6001)
341      WRITE(IOT,6007)
342      6007 FORMAT(T4,'DETERMINE HEAT AND HOT FLOW REQUIREMENTS - EACH DUTY CY
343      CLE')
344      WRITE (IOT,672)
345      C
346      WRITE (IOT,627) (HTKO(J),J=1,KCYCLE)
347      627 FORMAT(T4,'HTKO =',T13,10(F8.3,4X))

```



```

***** FUELCL *****
348 WRITE (IOT,628) (HTKH(J),J=1,KCYCLE)
349 628 FORMAT(T4,'HTKH =',T13,10(F8.3,4X))
350 WRITE (IOT,629) (Q1ODTR(J),J=1,KCYCLE)
351 629 FORMAT(T4,'Q1ODTR=',T13,10(F8.1,4X))
352 WRITE (IOT,630) (Q1HDTR(J),J=1,KCYCLE)
353 630 FORMAT(T4,'Q1HDTR=',T13,10(F8.1,4X))
354 WRITE (IOT,631) (WDT1FO(J),J=1,KCYCLE)
355 631 FORMAT(T4,'WDT1FO=',T13,10(F8.3,4X))
356 WRITE (IOT,632) (WDT1FH(J),J=1,KCYCLE)
357 632 FORMAT(T4,'WDT1FH=',T13,10(F8.3,4X))
358 WRITE (IOT,633) (Q2ODTR(J),J=1,KCYCLE)
359 633 FORMAT(T4,'Q2ODTR=',T13,10(F8.1,4X))
360 WRITE (IOT,634) (Q2HDTR(J),J=1,KCYCLE)
361 634 FORMAT(T4,'Q2HDTR=',T13,10(F8.1,4X))
362 WRITE (IOT,635) (WDT2FO(J),J=1,KCYCLE)
363 635 FORMAT(T4,'WDT2FO=',T13,10(F8.3,4X))
364 WRITE (IOT,636) (WDT2FH(J),J=1,KCYCLE)
365 636 FORMAT(T4,'WDT2FH=',T13,10(F8.3,4X))
366 C
367 WRITE (IOT,672)
368 C
369 WRITE (IOT,6008)
370 6008 FORMAT(T4,'DETERMINE ENERGY DERIVATIVE -EACH CYCLE')
371 WRITE (IOT,672)
372 C
373 WRITE (IOT,637) (CSBVFO(J),J=1,KCYCLE)
374 637 FORMAT(T4,'CSBVFO=',T13,10(F8.3,4X))
375 WRITE (IOT,638) (CSBVFH(J),J=1,KCYCLE)
376 638 FORMAT(T4,'CSBVFH=',T13,10(F8.3,4X))
377 WRITE (IOT,639) (PHIFO2(J),J=1,KCYCLE)
378 639 FORMAT(T4,'PHIFO2=',T13,10(F8.3,4X))
379 WRITE (IOT,640) (PHIFH2(J),J=1,KCYCLE)
380 640 FORMAT(T4,'PHIFH2=',T13,10(F8.3,4X))
381 C
382 C *****
383 C
384 C *** CHECK TO ASSURE ADEQUATE SUPPLY OF FUEL CELL REJECT HEAT
385 C
386 QTOTR = 0.0
387 DO 500 I = 1,KCYCLE
388 QSUMR(I) = Q1ODTR(I) + Q1HDTR(I) + Q2ODTR(I) + Q2HDTR(I)
389 QGANET(I) = QAVAIL(I) - QSUMR(I)
390 QTOTR = QTOTR + QSUMR(I)
391 500 CONTINUE
392 C
393 QEXCES = QFCOT - QTOTR
394 C
395 C *** COMPUTE MAX FLOW RATE OF F21 COOLANT REQUIRED.
396 C
397 WF2IMX = 0.0
398 DO 501 I = 1,KCYCLE
399 IF (WDT1FO(I).LT.WF2IMX) GO TO 502
400 WF2IMX = WDT1FO(I)
401 502 IF (WDT1FH(I).LT.WF2IMX) GO TO 503
402 WF2IMX = WDT1FH(I)
403 503 IF (WDT2FO(I).LT.WF2IMX) GO TO 504
404 WF2IMX = WDT2FO(I)
405 504 IF (WDT2FH(I).LT.WF2IMX) GO TO 501

```

```

***** FUELCL *****
406      WF2IMX = WD2FH(1)
407      501 CONTINUE
408      C
409      DGAMIN = DGANET(1)
410      DO 505 I = 2,KCYCLE
411      IF(DGANET(I).EQ.0.0) GO TO 505
412      DGAMIN = AMIN(DGAMIN,DGANET(I))
413      505 CONTINUE
414      C
415      C
416      IF(DGAMIN) 506,506,507
417      C
418      506 CONTINUE
419      C
420      507 CONTINUE
421      C
422      WRITE (107,672)
423      WRITE (107,6009)
424      6009 FORMAT(T4,'CHECK ADEQUACY OF REJECTED HEAT SUPPLY')
425      WRITE (107,672)
426      C
427      WRITE (107,641) (QSUMR(J),J=1,KCYCLE)
428      641 FORMAT(T4,'QSUMR =',T13,10(F8.1,4X))
429      WRITE (107,642) (DGANET(J),J=1,KCYCLE)
430      642 FORMAT(T4,'DGANET =',T13,10(F10.0,2X))
431      WRITE (107,643) QTOTR, QEXCES
432      643 FORMAT(T4,'QTOTR =',T13,F8.1,T25,'QEXCES =',T13,F12.0)
433      WRITE (107,644) WF2IMX, DGAMIN
434      644 FORMAT(T4,'WF2IMX =',T13,F8.2,T25,'DGAMIN =',T13,F12.0)
435      C
436      C *****
437      C
438      C *** COMPUTE THE WEIGHT OF PROPELLANT TANK HEATER CIRCULATING
439      C * COMPRESSOR. FIRST COMPUTE THE MAXIMUM HEAT FLOW RATE
440      C *** REQUIRED INTO THE TANK
441      C
442      TKOMAX = 0.0
443      TKHMAX = 0.0
444      QMXTKO = 0.0
445      QMXTKH = 0.0
446      DO 510 I = 1,KCYCLE
447      QMXTKO = AMAX1(QMXTKO,Q20DTR(I))
448      QMXTKH = AMAX1(QMXTKH,Q2HDTR(I))
449      TKOMAX = AMAX1(TKOMAX,TKO(I))
450      TKHMAX = AMAX1(TKHMAX,TKH(I))
451      510 CONTINUE
452      C
453      C *** COMPUTE THE SPECIFIC HEAT AND DENSITY OF H2 AND O2 AT
454      C * THE FINAL TEMPERATURE AND PRESSURE
455      C
456      PRFCMN(1) = PFOFC
457      PRFCMN(2) = PFHFC
458      C
459      CALL DENSON(TKOMAX,PRFCMN(1),1,RHOFIN(1),ZEE)
460      CALL GSDNST(2,TKHMAX,PRFCMN(2),RHOFIN(2))
461      CALL CSUBP(TKOMAX,PRFCMN(1),1,CPFO)
462      CALL CSUBP(TKHMAX,PRFCMN(2),2,CPFH)
463      C

```

```

***** FUELCL *****
464      WDTCF0 = GMXTKO/(CPFO * (TF2IIN - 100. - TKOMAX))
465      C
466      WDTCFH = GMXTKH/(CPFH * (TF2IIN - 125. - TKHMAX))
467      C
468      WOCMP = (0.01455 * DELTCP * WDTCF0)/RHOFIN(1)
469      WCIRCP(1) = WOCMP
470      WHCMP = (0.01455 * DELTCP * WDTCFH)/RHOFIN(2)
471      WCIRCP(2) = WHCMP
472      C
473      WRITE (IOT,672)
474      WRITE(IOT,6010)
475      6010 FORMAT(T4,'COMPUTE WEIGHT OF CIRCULATING PUMPS FOR TANKS')
476      WRITE (IOT,672)
477      C
478      WRITE (IOT,645) TKOMAX, TKHMAX
479      645 FORMAT(T4,'TKOMAX=',T13,F8.2,T25,'TKHMAX=',T13,F8.2)
480      WRITE (IOT,646) GMXTKO, GMXTKH
481      646 FORMAT(T4,'GMXTKO=',T13,F8.1,T25,'GMXTKH=',T13,F8.1)
482      WRITE (IOT,647) RHOFIN(1), RHOFIN(2)
483      647 FORMAT(T4,'RHOFIN-02=',T15,F8.3,T30,'RHOFIN-H2=',T14,F8.3)
484      WRITE (IOT,648) CPFO, CPHH
485      648 FORMAT(T4,'CPFO =',T13,F8.3,T25,'CPFH =',T13,F8.3)
486      WRITE (IOT,649) WDTCF0, WDTCFH
487      649 FORMAT(T4,'WDTCF0=',T13,F8.2,T25,'WDTCFH=',T13,F8.2)
488      WRITE (IOT,650) WOCMP, WHCMP
489      650 FORMAT(T4,'WOCMP =',T13,F8.2,T25,'WHCMP =',T13,F8.2)
490      C
491      C *****
492      C
493      C *** COMPUTE RESERVE REACTANT QUANTITY ***
494      C
495      POWMAX = 0.0
496      I = 0
497      DO 20 II = 1,NDCYCL,2
498      I = I + 1
499      POWMAX = POWMAX + PKHMAX * DCYCLE(II) * NEOP(I)
500      20 CONTINUE
501      C
502      C * SET RESERVE AT 11.5 PERCENT OF MISSION EXTRAPOLATED MAX
503      C POWER REACTANT REQUIREMENT *
504      C
505      WRMX = SRCFC * POWMAX
506      C
507      WRRSRV = SRCFC * POWMAX * 0.115
508      C
509      C
510      WORSRV = WRRSRV * C1
511      WRRSPV = WRRSRV * C2
512      C
513      C *** COMPUTE WEIGHT OF RESIDUAL REACTANTS ***
514      C
515      C
516      WTRES(1) = (RHOFIN(1)/RHOFIL(1))*(1.0/(1.0-(RHOFIN(1)/RHOFIL(1))))
517      I * (WOCNS + WORSRV + WOVNT)
518      C
519      WTRES(2) = (RHOFIN(2)/RHOFIL(2))*(1.0/(1.0-(RHOFIN(2)/RHOFIL(2))))
520      I * (WHCONS + WRRSRV + WOVNT)
521      C

```

0 LBS.
0 LBS.

```

***** FUELCL *****
522 C
523 C *** COMPUTE WEIGHT OF PURGE REACTANTS FOR FUEL CELLS.
524 C
525 AMPHRS = (POHTOT * 1000.0)/FCVOLT
526 C
527 PURGAS(1) = PRGRAT(1) * PRGTIM(1) * (AMPHRS/PRGINT(1))
528 C
529 PURGAS(2) = PRGRAT(2) * PRGTIM(2) * (AMPHRS/PRGINT(2))
530 C
531 NPRGE1 = AMPHRS/PRGINT(1)
532 IF (PRGINT(1).EQ.0.0) NPRGE1 = 0
533 NPRGE2 = AMPHRS/PRGINT(2)
534 IF (PRGINT(2).EQ.0.0) NPRGE2 = 0
535 C
536 JP = PAGE(0)
537 WRITE (IOT,6116)
538 WRITE (IOT,6001)
539 WRITE (IOT,6012)
540 6012 FORMAT(T4,'COMPUTE RESERVE, RESIDUAL AND PURGE REACTANT QUANTITIES
541 1')
542 WRITE (IOT,672)
543 WRITE (IOT,651) POWMAX, WRMAX
544 651 FORMAT(T4,'POWMAX=' ,T13,F8.1,T25,'WRMAX=' ,T37,F8.2)
545 WRITE (IOT,652) WRRSRV, WORSRV, WHRSRV
546 652 FORMAT(T4,'WRRSRV=' ,T13,F8.2,T25,'WORSRV=' ,T37,F8.2,T48,'WHRSRV=' ,
547 1 T60,F8.2)
548 WRITE (IOT,654) WTRES(1), WTRES(2)
549 654 FORMAT(T4,'WTRES-O2=' ,T17,F7.2,T25,'WTRES-H2=' ,T37,F7.2)
550 WRITE (IOT,682) AMPHRS, NPRGE1, NPRGE2
551 682 FORMAT(T4,'TOTAL AMPERE-HOURS=' ,F10.2,T40,'NUMBER OF FUEL CELL PU-
552 RGES - FOR OXYGEN SIDE IS' ,I4,' - FOR HYDROGEN SIDE IS' ,I4)
553 WRITE (IOT,683) PURGAS(1), PURGAS(2)
554 683 FORMAT(T4,'QUANTITY OF PURGE GAS USED IS - OXYGEN' ,F6.2,' - HYD
555 ROGEN' ,F6.2)
556 C
557 C *** COMPUTE VOLUME OF THE REACTANT TANKS.
558 C
559 VOLTNK(1) = (WOCONS + WORSRV + WOVENT + WTRES(1) + PURGAS(1))/(.97
560 1 * (RHOFIL(1) - RHOFIN(1)))
561 C
562 VOLTNK(2) = (WHCONS + WHRSRV + WHVENT + WTRES(2) + PURGAS(2))/(.98
563 1 * (RHOFIL(2) - RHOFIN(2)))
564 C
565 C *** COMPUTE AREA OF SPHERICAL REACTANT TANKS.
566 C
567 AREATK(1) = 4.84 * (VOLTNK(1)**0.666)
568 C
569 AREATK(2) = 4.84 * (VOLTNK(2)**0.666)
570 C
571 WRITE (IOT,672)
572 WRITE (IOT,6011)
573 6011 FORMAT(T4,'COMPUTE TANK VOLUME AND SURFACE AREA')
574 WRITE (IOT,672)
575 C
576 WRITE (IOT,655) VOLTNK(1), VOLTNK(2)
577 655 FORMAT(T4,'VOLTNK-O2=' ,T15,F8.2,T30,'VOLTNK-H2=' ,T42,F8.2)
578 WRITE (IOT,656) AREATK(1), AREATK(2)
579 656 FORMAT(T4,'AREATK-O2=' ,T15,F8.2,T30,'AREATK-H2=' ,T42,F8.2)

```

```

***** FUELCL *****
580 C
581 C *** COMPUTE HEAT LEAK INTO REACTANT TANKS, PER HOUR AND TOTAL
582 C
583 GLEAKO = 0.0
584 GLEAKH = 0.0
585 I = 0
586 DO 30 I1 = 2, NDCYCL, 2
587 I = I + 1
588 CALL TCOND(TENV, TKO(I), SNBAR(1), SITHIK(1,1), SITYPE(1,1), TOCOND)
589 QLK0(I) = TOCOND * AREATK(1) * DCYCLE(I1)
590 GLEAKO = GLEAKO + QLK0(I)
591 C
592 CALL TCOND(TENV, TKH(I), SNBAR(2), SITHIK(2,1), SITYPE(2,1), THCOND)
593 QLKH(I) = THCOND * AREATK(2) * DCYCLE(I1)
594 GLEAKH = GLEAKH + QLKH(I)
595 C
596 30 CONTINUE
597 C
598 C *****
599 C
600 C
601 C BEGIN CALCULATING WEIGHT OF H2 VENTED DURING THE MISSION. FIRST
602 C COMPUTE THE SPECIFIC HEAT OF H2 AT T= TKH(I) AND P= PCHFC
603 C
604 WVHO = 0.0
605 WVHH = 0.0
606 PVENTO = SVPRES(1,1)
607 PVENTH = SVPRES(2,1)
608 C
609 DO 550 I = 1, KCYCLE
610 C
611 CSBV02 = CSUBV(TKO(I), PCOFC, 1)
612 CSBVH2 = CSUBV(TKH(I), PCHFC, 2)
613 GRQDO2 = ((VOLTNK(1) * CSBV02) / 48.3) * (PVENTO - PCOFC) * 144.0
614 GRQDH2 = ((VOLTNK(2) * CSBVH2) / 776.5) * (PVENTH - PCHFC) * 144.0
615 IF (GRQDO2.GT. QLK0(I)) GO TO 31
616 DELQ = QLK0(I) - GRQDO2
617 WVENTO = DELQ / (CSBV02 * TKO(I) * ((PVENTO / PCOFC) - 1.0))
618 31 WVHO = WVHO + WVENTO
619 IF (GRQDH2.GT. QLKH(I)) GO TO 32
620 DELQH = QLKH(I) - GRQDH2
621 WVENTH = DELQH / (CSBVH2 * TKH(I) * ((PVENTH / PCHFC) - 1.0))
622 32 WVHH = WVHH + WVENTH
623 C
624 550 CONTINUE
625 C
626 C COMPUTE RESULTANT TOTAL WEIGHT OF VENTED GASES.
627 C
628 IF (WVHO.LT.0.0) WVHO = 0.0
629 IF (WVHH.LT.0.0) WVHH = 0.0
630 C
631 WOVENT = WVHO
632 WHVENT = WVHH
633 C
634 WRITE (IOT, 672)
635 WRITE (IOT, 6013)
636 6013 FORMAT (T4, 'COMPUTE HEAT LEAK AND VENTED REACTANT QUANTITY')
637 WRITE (IOT, 672)

```

***** FUELCL *****

```

638 C
639 WRITE (10T,657) (QLKO(J),J=1,KCYCLE)
640 657 FORMAT(T4,'QLKO =',T13,10(E9.4,3X))
641 WRITE (10T,658) (QLKH(J),J=1,KCYCLE)
642 658 FORMAT(T4,'OLKH =',T13,10(E9.4,3X))
643 WRITE (10T,659) QLEAKO, QLEAKH
644 659 FOPMAT(T4,'QLEAKO=',T13,F10.5,T30,'QLEAKH=',T39,E10.9)
645 WRITE (10T,660) WVHO, WVHH
646 660 FORMAT(T4,'WVHO =',T13,F8.2,T25,'WVHH =',T37,F8.2)
647 WRITE (10T,661) WOVENT, WHVENT
648 661 FORMAT(T4,'WOVENT=',T13,F8.2,T25,'WHVENT=',T37,F8.2)
649 C
650 C *****
651 C
652 C *** COMPUTE REACTANT LOADED INTO TANKS
653 C
654 WRTOTL(1) = (WOCONS + WORSRV + WOVENT + WTRES(1) + PURGAS(1))
655 C
656 WRTOTL(2) = (WHCONS + WHRSRV + WHVENT + WTRES(2) + PURGAS(2))
657 C
658 C *** COMPUTE DIAMETER OF REACTANT TANKS - ASSUMED SPHERICAL
659 C
660 DIATK(1) = ((1.9098 * (WRTOTL(1)/RHOFIL(1)))**0.33) * 12.0
661 DIATK(2) = ((1.9098 * (WRTOTL(2)/RHOFIL(2)))**0.33) * 12.0
662 C
663 ITANK1 = 0
664 IF(DIATK(1).GT.TKMXDI(1)) ITANK1 = 1
665 ITANK2 = 0
666 IF(DIATK(2).GT.TKMXDI(2)) ITANK2 = 1
667 C
668 C *** COMPUTE REACTANT TANK INSULATION WEIGHT
669 C
670 IT1 = SITYPE(1,1)
671 IT2 = SITYPE(2,1)
672 C
673 TIWT(1,1) = NOP(1,1) * AREATK(1) * RHOI(IT1) * SITHIK(1,1)/12.0
674 TIWT(2,1) = NOP(2,1) * AREATK(2) * RHOI(IT2) * SITHIK(2,1)/12.0
675 C
676 C *** COMPUTE DIAMETER OF REACTANT TANK VACUUM JACKETS
677 C
678 DIAVJ(1) = DIATK(1) + (VJANUL(1) * 2.0)
679 DIAVJ(2) = DIATK(2) + (VJANUL(2) * 2.0)
680 C
681 C *** COMPUTE WEIGHT OF REACTANT TANK PRESSURE VESSELS
682 C
683 MATL1 = SHTYPE(1,1)
684 CALL FINTAD(MTBID(9),MATL1)
685 FTUX1 = MIPE(1,TKHMAX)
686 C
687 THKMT1 = (1.0 * PCOFC * 2.0 * (DIATK(1)/2.0))/FTUX1
688 IF(THKMT1.LT.MINTHK(MATL1)) THKMT1 = MINTHK(MATL1)
689 C
690 MATL2 = SHTYPE(2,1)
691 CALL FINTAD(MTBID(9),MATL2)
692 FTUX2 = MIPE(1,TKHMAX)
693 C
694 THKMT2 = (1.0 * PCFC * 2.0 * (DIATK(2)/2.0))/FTUX2
695 IF(THKMT2.LT.MINTHK(MATL2)) THKMT2 = MINTHK(MATL2)

```

```

***** FUELCL *****
696 C
697 WTPVT(1) = 1.35 * AREATK(1) * RHOL(MATL1) * (THKMT1/12.0)
698 C
699 WTPVT(2) = 1.35 * AREATK(2) * RHOL(MATL2) * (THKMT2/12.0)
700 C
701 *** COMPUTE WEIGHT OF VACUUM JACKET FOR REACTANT TANKS
702 C ASSUME SPHERICAL VACUUM JACKET (HARD SHELL) 6061-T6 ALUMINUM
703 C
704 MATL3 = 13
705 CALL FINTAB (NTBID(9)+MATL3)
706 FTUX3 = MIPE(1,TEHV)
707 C
708 THKMT3 = (1.0 * (PCOFC/2.0) * (DIAVJ(1)/2.0))/FTUX3
709 IF (THKMT3.LT.MINTHK(MATL3)) THKMT3 = MINTHK(MATL3)
710 C
711 THKMT4 = (1.0 * (PCHFC/2.0) * (DIAVJ(2)/2.0))/FTUX3
712 IF (THKMT4.LT.MINTHK(MATL3)) THKMT4 = MINTHK(MATL3)
713 C
714 WTVJ(1) = (PI * (DIAVJ(1)**2)/144.0) * RHOL(MATL3)*(THKMT3/12.0)
715 C
716 WTVJ(2) = (PI * (DIAVJ(2)**2)/144.0) * RHOL(MATL3)*(THKMT4/12.0)
717 C
718 *** COMPUTE TOTAL WEIGHT OF TANK
719 C
720 WTTOT(1) = WTPVT(1) + WTVJ(1) + TIWT(1,1)
721 WTTOT(2) = WTPVT(2) + WTVJ(2) + TIWT(2,1)
722 C
723 WRITE (IOT,672)
724 WRITE (IOT,6014)
725 6014 FORMAT(T4,'COMPUTE PRESSURE VESSEL, VACUUM JACKET, INSULATION AND
726 TOTAL TANK WEIGHTS')
727 WRITE (IOT,672)
728 C
729 WRITE (IOT,662) WTTOTL(1), WTTOTL(2)
730 662 FORMAT(T4,'WTTOTL-02 =',T15,F8.2,T30,'WTTOTL-H2 =',T42,F8.2)
731 WRITE (IOT,663) DIATK(1), DIATK(2)
732 663 FORMAT(T4,'DIATK-02 =',T15,F8.2,T30,'DIATK-H2 =',T42,F8.2)
733 C
734 IF (ITANK1.EQ.1) WRITE (IOT,680) DIATK(1),TKMXDI(1)
735 680 FORMAT( ' *** O2 STORAGE TANK DIAMETER ',F6.2,' INCHES EXCEEDS TH
736 IE MAXIMUM ALLOWABLE PRESSURE VESSEL DIAMETER OF ',F6.2,' INCHES, ',/
737 :2 T5,' ADDITIONAL TANKAGE MUST BE PROVIDED. ***')
738 C
739 IF (ITANK2.EQ.1) WRITE (IOT,681) DIATK(2),TKMXDI(2)
740 681 FORMAT( ' *** H2 STORAGE TANK DIAMETER ',F6.2,' INCHES EXCEEDS TH
741 IE MAXIMUM ALLOWABLE PRESSURE VESSEL DIAMETER OF ',F6.2,' INCHES, ',/
742 :2 T5,' ADDITIONAL TANKAGE MUST BE PROVIDED. ***')
743 C
744 WRITE (IOT,664) TIWT(1,1), TIWT(2,1)
745 664 FORMAT(T4,'TIWT-02 =',T15,F8.2,T30,'TIWT-H2 =',T42,F8.2)
746 WRITE (IOT,665) DIAVJ(1), DIAVJ(2)
747 665 FORMAT(T4,'DIAVJ-02 =',T15,F8.2,T30,'DIAVJ-H2 =',T42,F8.2)
748 WRITE (IOT,666) RHOFTU(1), RHOFTU(2)
749 666 FORMAT(T4,'RHOFTU-02 =',T15,F8.7,T30,'RHOFTU-H2 =',T42,F8.7)
750 WRITE (IOT,667) WTPVT(1), WTPVT(2)
751 667 FORMAT(T4,'WTPVT-02 =',T15,F8.2,T30,'WTPVT-H2 =',T42,F8.2)
752 WRITE (IOT,668) WTVJ(1), WTVJ(2)
753 668 FORMAT(T4,'WTVJ-02 =',T15,F8.2,T30,'WTVJ-H2 =',T42,F8.2)

```

```

***** FUELCL *****
754      WRITE (IOT,669) WTTOT(1), WTTOT(2)
755      669 FORMAT(T4, 'WTTOT-02 =', T15, F8.2, T30, 'WTTOT-M2 =', T42, F8.2)
756      C
757      C *****
758      C
759      C      *** COMPUTE WEIGHT OF FUEL CELLS
760      C
761      FCWGT = PKHMAX * SPWTF * (NFCOP + NFCSTB)
762      ENGWT = FCWGT
763      C
764      WRITE (IOT,672)
765      WRITE (IOT,6015) NFCOP, NFCSTB
766      6015 FORMAT(T4, 'COMPUTE WEIGHT OF FUEL CELLS', T15, I3, 2X, 'OPERATING AND',
767      1, I3, 2X, 'STANDBY')
768      WRITE (IOT,672)
769      C
770      WRITE (IOT,670) FCWGT
771      670 FORMAT(T4, 'FCWGT =', T15, F8.2)
772      C
773      C *****
774      C
775      C      *** COMPUTE WEIGHT OF HEAT EXCHANGERS
776      C      * FIRST COMPUTE THE HOT FLUID TEMPERATURE DROP
777      C      *** IN THE SERIES HEAT EXCHANGER SEQUENCE
778      C
779      DTO1MX = 0.0
780      DTH1MX = 0.0
781      DTO2MX = 0.0
782      DTH2MX = 0.0
783      C
784      DO 100 I = 1, KCYCLE
785      DTO1MX = AMAX1(DTO1MX, WDT1FO(I))
786      DTH1MX = AMAX1(DTH1MX, WDT1FH(I))
787      DTO2MX = AMAX1(DTO2MX, WDT2FO(I))
788      DTH2MX = AMAX1(DTH2MX, WDT2FH(I))
789      100 CONTINUE
790      IF (DTO1MX, LT, DTO2MX) DTO1MX = DTO2MX
791      IF (DTH1MX, LT, DTH2MX) DTH1MX = DTH2MX
792      IF (DTO1MX, LT, DTH1MX) DTO1MX = DTH1MX
793      WDTFMX = DTO1MX
794      C
795      C
796      Q10DMX = 0.0
797      DO 110 I = 1, KCYCLE
798      IF (Q10DTR(I), LT, Q10DMX) GO TO 110
799      Q10DMX = Q10DTR(I)
800      IMAX0 = I
801      110 CONTINUE
802      C
803      Q1HDMX = 0.0
804      DO 120 I = 1, KCYCLE
805      IF (Q1HDTR(I), LT, Q1HDMX) GO TO 120
806      Q1HDMX = Q1HDTR(I)
807      IMAXH = I
808      120 CONTINUE
809      C
810      T2F21 = TF21IN - (Q10DTR(IMAX0) / (CF21 * WDTFMX))
811      C

```



```

***** FUELCL *****
812      T3F21 = T2F21 - (GMXTKO/(CF21 * WDTFMX))
813      T4F21 = T3F21 - (Q1HDTX(IMAXH)/(CF21 * WDTFMX))
814      C
815      T5F21 = T4F21 - (GMXTKH/(CF21 * WDTFMX))
816      C
817      JX = 0
818      QTREQ(1) = 0.0
819      QTREQ(2) = 0.0
820      C
821      *** COMPUTE WEIGHT AND CHARACTERISTICS OF O2 HEX BETWEEN
822      *
823      C
824      JX = JX + 1
825      IGAS = 1
826      C
827      WDOTX(JX,IGAS) = WDTIFO(IMAXO)      B LBS PER HOUR
828      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
829      HEXHIT(JX,IGAS) = TF21IN
830      HEXCIT(JX,IGAS) = TKO(IMAXO)
831      HEXHOT(JX,IGAS) = T2F21
832      HEXCOT(JX,IGAS) = TFCNOM(1)
833      HEXCOP(JX,IGAS) = PCOFC - 3.0
834      HEXCIP(JX,IGAS) = PCOFC
835      C
836      HSOREQ(JX,IGAS) = Q1ODMX
837      QTREQ(IGAS) = QTREQ(IGAS) + HSOREQ(JX,IGAS)
838      CALL CSURP(TKO(IMAXO),PCOFC,1,CPCLDF(JX,IGAS))
839      TMF211 = (TF21IN + T2F21)/2.0
840      CPF211 = CSPF21(TM211)
841      CPHOTF(JX,IGAS) = CPF211
842      HSGCPE(JX,IGAS) = CPF211
843      HSGTOT(JX,IGAS) = Q1ODMX/(CPF211 * (TF21IN - T2F21))
844      C
845      WDOTH(JX,IGAS) = Q1ODMX/(CPF211 * (TF21IN - T2F21))
846      CALL HEXF21(IGAS,Q1ODMX,HEXCIP(JX,IGAS),WHXTOT(JX,IGAS))
847      C
848      C
849      C
850      *** COMPUTE WEIGHT AND CHARACTERISTICS OF H2 HEX BETWEEN
851      *
852      C
853      IGAS = 2
854      C
855      WDOTX(JX,IGAS) = WDTIFH(IMAXH)      B LBS PER HOUR
856      UCODE(JX,IGAS) = HXCODE(JX,IGAS)
857      HEXHIT(JX,IGAS) = T3F21
858      HEXCIT(JX,IGAS) = TKH(IMAXH)
859      HEXHOT(JX,IGAS) = T4F21
860      HEXCOT(JX,IGAS) = TFCNOM(2)
861      HEXCOP(JX,IGAS) = PCHFC - 3.0
862      HEXCIP(JX,IGAS) = PCHFC
863      C
864      HSOREQ(JX,IGAS) = Q1HDMX
865      QTREQ(IGAS) = QTREQ(IGAS) + HSOREQ(JX,IGAS)
866      CALL CSURP(TKH(IMAXH),PCHFC,2,CPCLDF(JX,IGAS))
867      TMF212 = (T3F21 + T4F21)/2.0
868      CPF212 = CSPF21(TM212)
869      CPHOTF(JX,IGAS) = CPF212

```

```

***** FUELCL *****
870 HSGCPE(JX,IGAS) = CPF212
871 HSGTOT(JX,IGAS) = Q1HDMX/(CPF212 * (T3F21 - T4F21))
872 WDOOTH(JX,IGAS) = Q1HDMX/(CPF212 * (T3F21 - T4F21))
873 C
874 CALL HEXF21(IGAS,Q1HDMX,HEXCIP(JX,IGAS),WHXTOT(JX,IGAS))
875 C
876 C
877 C *** COMPUTE WEIGHT AND CHARACTERISTICS OF O2 HEX BETWEEN
878 C * O2 TANK CIRCULATOR AND O2 TANK
879 C
880 JX = JX + 1
881 IGAS = 1
882 C
883 WDOTX(JX,IGAS) = WDTCPF0
884 UCODE(JX,IGAS) = HXCODE(JX,IGAS)
885 HEXHIT(JX,IGAS) = T2F21
886 HEXCIT(JX,IGAS) = TKO(1)
887 HEXHOT(JX,IGAS) = T3F21
888 HEXCOT(JX,IGAS) = TKOMAX
889 HEXCOP(JX,IGAS) = PCOFC
890 HEXCIP(JX,IGAS) = PCOFC + DELTCP
891 C
892 HSOREQ(JX,IGAS) = QMXTKO
893 QTREQ(IGAS) = QTREQ(IGAS) + HSOREQ(JX,IGAS)
894 CALL CSUBP(TKO(1),PCOFC,1,CPCLOF(JX,IGAS))
895 THF213 = (T2F21 + T3F21)/2.0
896 CPF213 = CSPPF21(THF213)
897 CPHOTF(JX,IGAS) = CPF213
898 HSGCPE(JX,IGAS) = CPF213
899 HSGTOT(JX,IGAS) = QMXTKO/(CPF213 * (T2F21 - T3F21))
900 WDOOTH(JX,IGAS) = QMXTKO/(CPF213 * (T2F21 - T3F21))
901 C
902 CALL HEXF21(IGAS,QMXTKO,HEXCIP(JX,IGAS),WHXTOT(JX,IGAS))
903 C
904 C
905 C *** COMPUTE WEIGHT AND CHARACTERISTICS OF H2 HEX BETWEEN
906 C * H2 TANK CIRCULATOR AND H2 TANK
907 C
908 IGAS = 2
909 C
910 WDOTX(JX,IGAS) = WDTCFH
911 UCODE(JX,IGAS) = HXCODE(JX,IGAS)
912 HEXHIT(JX,IGAS) = T4F21
913 HEXCIT(JX,IGAS) = TKH(1)
914 HEXHOT(JX,IGAS) = T5F21
915 HEXCOT(JX,IGAS) = TKHMAX
916 HEXCOP(JX,IGAS) = PCHFC
917 HEXCIP(JX,IGAS) = PCHFC + DELTCP
918 C
919 HSOREQ(JX,IGAS) = QMXTKH
920 QTREQ(IGAS) = QTREQ(IGAS) + HSOREQ(JX,IGAS)
921 CALL CSUBP(TKH(1),PCHFC,2,CPCLOF(JX,IGAS))
922 THF214 = (T4F21 + T5F21)/2.0
923 CPF214 = CSPPF21(THF214)
924 CPHOTF(JX,IGAS) = CPF214
925 HSGCPE(JX,IGAS) = CPF214
926 HSGTOT(JX,IGAS) = QMXTKH/(CPF214 * (T4F21 - T5F21))
927 WDOOTH(JX,IGAS) = QMXTKH/(CPF214 * (T4F21 - T5F21))

```

```

***** FUELCL *****
928 C
929 CALL HEXF2(IGAS,QMXTKH,HEXCIP(JX,IGAS),WHXTOT(JX,IGAS))
930 C
931 C
932 CALL OTHXF
933 C
934 C
935 C *****
936 C
937 C *** COMPUTE THE TANK ENERGY HISTORY AND HEATER DUTY CYCLE ***
938 C
939 TIMINC = 1.0 B PERIOD DEFINED BY GAS FLOW
940 PSET1 = PLSET1
941 PSET2 = PLSET2
942 PTANK1 = PCOFC
943 PTANK2 = PCHFC
944 C
945 LI = 0
946 DO 200 KI = 1,NDCYCL*2
947 LI = LI + 1
948 TIM(LI) = DCYCLE(KI)
949 200 CONTINUE
950 C
951 TIME = 0.0
952 ICHTO = 0
953 ICHTH = 0
954 TKOW = 0.0
955 TKHW = 0.0
956 QICUM = 0.0
957 Q2CUM = 0.0
958 C
959 DO 210 K = 1,KCYCLE
960 C
961 CTIM = TIM(K) * 60.0
962 IF(CTIM.LT.10.0) GO TO 211
963 NTP = (TIM(K) * 6.0) + 0.6
964 WDT030 = WDTFCO(K)/6.0
965 WDT030 = WDTFCH(K)/6.0
966 GO TO 212
967 C
968 211 NTP = CTIM + 0.6
969 WDT030 = WDTFCO(K)/60.0
970 WDT030 = WDTFCH(K)/60.0
971 C
972 212 CONTINUE
973 JP = PAGE(0)
974 JP = PAGE(10)
975 C
976 WRITE (IOT,700)
977 700 FORMAT(/T42, '*** COMPUTE TANK ENERGY HISTORY AND HEATER DUTY CYCLE'
978 1 '***'/)
979 C
980 WRITE (IOT,701) K, TIM(K)
981 701 FORMAT(T5, 'SYSTEM DUTY CYCLE NUMBER IS ',I4,T40, 'CYCLE IS ',F6.2,
982 1 ' HOURS IN DURATION')
983 C
984 WRITE (IOT,702) K
985 702 FORMAT(T120, 'CYCLE ',I4,T4, 'TIME',T11, 'GAS',T20, 'PER-CENT',T33, 'D

```

```

***** FUELCL *****
986      DENSITY, T44, FLUID, T53, THETA, T63, PHI, T71, Q-REGD, T81, Q-CUM
987      2, T91, TANK, T100, Q-HTR, T110, HEATER, T11, FLOW, T20, WITHDRAWN
988      3, T44, TEMP, T91, PRES, T100, REGD, T110, TIME-ON, T121, HEATER
989      4, T3, MIN, T11, LBS, T20, PERCENT, T33, LB/CF, T44, R,
990      5, T53, R/LB, T61, P-CF/D, T72, BTU, T81, BTU, T91, PSIA,
991      6, T100, BTU, T110, MIN, T121, CYCLE, T11, O2/H2, T22, O2/H2,
992      T734, O2/H2, T44, O2/H2, T53, O2/H2, T62, O2/H2, T72, O2/H2, T81,
993      8, O2/H2, T91, O2/H2, T100, O2/H2, T110, O2/H2, T121, O2/H2,
994      C
995      C
996      DO 220 I = 1, NTP
997      C
998      * SET TIME COUNTERS TO APPROPRIATE TIME SUB-INTERVAL *
999      C
1000     IF (1.GT.1) GO TO 226
1001     PTIME = CTIM + TIME
1002     226 CONTINUE
1003     C
1004     DTIME = PTIME - TIME
1005     IK2 = DTIME
1006     IF (CTIM.LT.10.0) GO TO 225
1007     IF (DTIME.LT.10.0) GO TO 221
1008     IK2 = 0
1009     TIME = TIME + 10.0
1010     GO TO 222
1011     C
1012     224 DTIME = PTIME - TIME
1013     221 TIME = TIME + 1.0
1014     IK2 = IK2 + 1
1015     WDT030 = WDTFCO(K)/60.0
1016     WDTM30 = WDTFCH(K)/60.0
1017     GO TO 222
1018     C
1019     225 TIME = TIME + 1.0
1020     IK2 = 0
1021     C
1022     C
1023     222 CONTINUE
1024     TKOW = TKOW + WDT030
1025     TKODP = TKOW
1026     TKHW = TKHW + WDTM30
1027     TKHDP = TKHW
1028     C
1029     PCOXW = TKODP/(WCONSD + 0.2 * WCONSD)
1030     PCH2W = TKHDP/(WHCONS + 0.2 * WHCONS)
1031     C
1032     C
1033     C3 = 1.0 - ((0.0427*PFHFC)/(ZFH*TFHFC))
1034     C4 = 1.0 - ((0.04253*PFOFC)/(ZFO*TFOFC))
1035     C
1036     ORHO = 70.126 * (1.0 - (PCOXW * C4))
1037     HRHO = 4.365 * (1.0 - (PCH2W * C3))
1038     C
1039     CALL FINTAB (NTBID(8))
1040     XTAB(1) = PTANKI
1041     XTAB(2) = ORHO
1042     OXTEM = MIPE(2, XTAB)
1043     C

```

```

***** FUELCL *****
1044      CALL FINTAB (NTBID( 7))
1045      XTAB(1) = PTANK2
1046      XTAB(2) = HRHO
1047      H2TEM = MIPE(2,XTAB)
1048      C
1049      C
1050      CALL PHTON(OTEM,ORHO,1,PHI,THETA)
1051      DQDM1 = THETA
1052      DPDU1 = PHI
1053      C
1054      CALL FINTAB (NTBID(5))
1055      XTAB(1) = PTANK2
1056      XTAB(2) = HRHO
1057      DQDM2 = MIPE(2,XTAB)
1058      C
1059      CALL FINTAB (NTBID(40))
1060      XTAB(1) = PTANK2
1061      XTAB(2) = HRHO
1062      DPDU2 = MIPE(2,XTAB)
1063      C
1064      C
1065      QDTTK1 = WDT030 * DQDM1
1066      QDTTK2 = WDT030 * DQDM2
1067      Q1CUM = Q1CUM + QDTTK1
1068      Q2CUM = Q2CUM + QDTTK2
1069      C
1070      CALL BETAB(OTEM,ORHO,1,BETA0)
1071      C
1072      CALL FINTAB (NTBID(46))
1073      XTAB(1) = PTANK2
1074      XTAB(2) = H2TEM
1075      BETAH = MIPE(2,XTAB)
1076      C
1077      CALL CSUBP(OTEM,PTANK1,1,CPO)
1078      CALL CSUBP(H2TEM,PTANK2,2,CPH2)
1079      C
1080      C
1081      DELP1 = TIMINC * ((DPDU1/VOLTNK(1)) * ((-CPO/BETA0)*WDT030))
1082      DELP2 = TIMINC * ((DPDU2/VOLTNK(2)) * ((-CPH2/BETAH) * WDT030))
1083      C
1084      PTANK2 = PTANK2 + DELP2
1085      PTANK1 = PTANK1 + DELP1
1086      C
1087      QHSF21 = ((QDTEC * PKW(K) * NEOP(K))/60.0) * 0.75
1088      C
1089      IF(PTANK1.LE.PSET1) GO TO 130
1090      GO TO 135
1091      130 QTANK1 = Q1CUM
1092      HTRON1 = QTANK1/QHSF21
1093      ICHT0 = ICHT0 + 1
1094      135 IF(PTANK2.LE.PSET2) GO TO 140
1095      GO TO 118
1096      140 QTANK2 = Q2CUM
1097      HTRON2 = QTANK2/QHSF21
1098      ICHTH = ICHTH + 1
1099      C
1100      118 CONTINUE
1101      C

```

```

***** FUELCL *****
1102 IF (PAGE(2)) GO TO 227
1103 GO TO 228
1104 227 WRITE (IOT,702) K
1105 JP = PAGE(6)
1106 228 CONTINUE
1107 C
1108 WRITE (IOT,703) TIME,WDT030 ,PCOXW,ORHO,OXTEM,DGDM1,DPDU1,GDTTK1,
1109 1 GICUM,PTANK1,QTANK1,HTRON1,ICHT0,WDTH30,PCH2W,HRHO,H2TEM,DGDM2,
1110 2 OPDU2,GDTTK2,G2CUM ,PTANK2,QTANK2,HTRON2,ICHTH
1111 703 FORMAT(T1,F9.2,T11,F6.3,T21,F6.5,T32,F8.5,T43,F6.2,T52,F6.2,T61,
1112 1 F6.2,T71,F6.1,T80,F7.0,T90,F7.2,T99,F7.0,T109,F7.2,T120,16/T11,
1113 2 F6.3,T21,F6.5,T32,F8.5,T43,F6.2,T52,F6.2,T61,F6.2,T71,F6.1,T80,
1114 3 F7.0,T90,F7.2,T99,F7.0,T109,F7.2,T120,16)
1115 C
1116 IF (PTANK1,LE,PSET1) HTRON1 = 0.0
1117 IF (PTANK1,LE,PSET1) QTANK1 = 0.0
1118 IF (PTANK1,LE,PSET1) GICUM = 0.0
1119 IF (PTANK1,LE,PSET1) PTANK1 = PCOFC
1120 IF (PTANK2,LE,PSET2) HTRON2 = 0.0
1121 IF (PTANK2,LE,PSET2) QTANK2 = 0.0
1122 IF (PTANK2,LE,PSET2) G2CUM = 0.0
1123 IF (PTANK2,LE,PSET2) PTANK2 = PCHFC
1124 C
1125 IF (IK2,GE,1) GO TO 224
1126 C
1127 IF ((IK2,EQ,0).AND.(DTIME,LE,0.0)) GO TO 221
1128 C
1129 IF (CTIM,LT,10.0) GO TO 229
1130 C
1131 IF (DTIME,LT,10.0).AND.(DTIME,LE,0.0)) GO TO 224
1132 C
1133 229 CONTINUE
1134 C
1135 220 CONTINUE
1136 C
1137 221 CONTINUE
1138 C
1139 210 CONTINUE
1140 C
1141 *****
1142 C
1143 WRITE (IOT,6099)
1144 6099 FORMAT(//////T25,1***** THE SUPERCRITICAL FUEL CELL CALCULATIONS HA
1145 IVE BEEN COMPLETED *****)
1146 C
1147 RETURN
1148 C
1149 END

```

SUBROUTINE GASGEN

```

1      SUBROUTINE GASGEN(JX,IGAS)
2      C
3      INCLUDE CDCYCL
4      INCLUDE CHEX
5      INCLUDE CHSORC
6      INCLUDE CTANK
7      C
8      C      ***** COMPUTE THE HEAT SOURCE PARAMETERS.
9      C
10     WGGFU(JX,IGAS) = 0.0
11     WGGFX(JX,IGAS) = 0.0
12     WGGSBT(JX,IGAS) = 0.0
13     HSGREQ(JX,IGAS) = 0.0
14     HSGTOT(JX,IGAS) = 0.0
15     HSGCPE(JX,IGAS) = 0.0
16     C
17     JHSTYP = HSTYPE(JX,IGAS)
18     C
19     GO TO (10,20,30), JHSTYP
20     C
21     C      ***** FOR GAS GENERATOR ONLY AS HEAT SOURCE
22     C
23     10 HSGSUM = 0.0
24     DO 40 I2 = 1, NDCYCL, 2
25     HSGSUM = HSGSUM + WDOTH(JX,IGAS) * DCYCLE(I2)
26     40 CONTINUE
27     C
28     WGGFX(JX,IGAS) = HSGSUM - WGGFU(JX,IGAS)
29     WGGSBT(JX,IGAS) = WGGSBT(JX,IGAS) + WGGFX(JX,IGAS)
30     WGGTOT(IGAS) = WGGTOT(IGAS) + WGGFX(JX,IGAS)
31     C
32     ATERM = 13.824204 - (0.0117823 * HSPRES(JX,IGAS)) + (1.8632927E-5 *
33     1 (HSPRES(JX,IGAS)**2)) - (1.108423E-8 * (HSPRES(JX,IGAS)**3))
34     C
35     BTERM = 7.9470262 - (1.035636198 * HSPRES(JX,IGAS)) + (6.4684644E-5 *
36     1 (HSPRES(JX,IGAS)**2)) - (3.7946E-8 * (HSPRES(JX,IGAS)**3))
37     C
38     HSWGHT(JX,IGAS) = ATERM + BTERM * WDOTH(JX,IGAS)
39     C
40     HXASSY(JX,IGAS) = WHXTOT(JX,IGAS) + HSWGHT(JX,IGAS)
41     C
42     RETURN
43     C
44     C      ***** FOR WASTE HEAT ONLY AS HEAT SOURCE.
45     C
46     20 HSGSUM = 0.0
47     DO 50 I2 = 1, NDCYCL, 2
48     HSGSUM = HSGSUM + WDOTH(JX,IGAS) * DCYCLE(I2)
49     50 CONTINUE
50     C
51     CALL CSUBP1(HSOTEN(JX,IGAS), HSHRAT(JX,IGAS), CPEG)
52     HSGREQ(JX,IGAS) = HSGSUM * CPEG * (HEXHIT(JX,IGAS) - HEXHOT(JX,IGAS))
53     QTRQ(IGAS) = QTRQ(IGAS) + HSGREQ(JX,IGAS)
54     HSGTOT(JX,IGAS) = HSGSUM
55     HFTOT(IGAS) = HFTOT(IGAS) + HSGTOT(JX,IGAS)
56     HSGCPE(JX,IGAS) = CPEG
57     C

```

***** .GASGEN *****

```
58      RETURN
59      C
60      C      ***** FOR GAS GENERATOR AND WASTE HEAT BOTH AS HEAT SOURCES.
61      C
62      30 WGGFU(JX,IGAS) = HSAEE(JX,IGAS)/((9.11625 + HSMRAT(JX,IGAS) *
63      1 (0.46875 + HSMRAT(JX,IGAS) - 1.675)) * (HEXHIT(JX,IGAS) -
64      2 HEXHOT(JX,IGAS)))
65      C
66      GO TO 10
67      C
68      END
```

SUBROUTINE GETCON

```

1      C      * * * * *
2      C      * ROUTINE NAME - ROUTINE TO UNPACK THE FIRST *
3      C      * WORD OF THE CONFIGURATION *
4      C      * TABE. SEE S.R. STDCON FOR *
5      C      * WORD FORMAT *
6      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
7      C      * PROGRAMMER - R. BOLLINGER 1943 102 26933 *
8      C      * DATE CODED - 3/11/70 *
9      C      * * * * *
10     C
11     C      SUBROUTINE GETCON(IDX)
12     C
13     C      ***** EXPLANATION OF THE CALLING SEQUENCE
14     C      *
15     C      ***** IDX - INDEX OF THE CONFIGURATION TABLE
16     C
17     C      LOGICAL DIAG
18     C
19     C      INCLUDE CCNFIG
20     C      INCLUDE CIOUNT
21     C
22     C      DO 10 I1 = 1,6
23     C      ICNFIG(I1) = IPBYTE(CONFIG(IDX,I1),I1,0,6)
24     C      10 CONTINUE
25     C
26     C      IF (DIAG(1,6HGETCON)) WRITE (10T,6000) IDX,ICNFIG
27     C      RETURN
28     C      6000 FORMAT (1,14X,7I5)
29     C      END

```

FUNCTION GOMTRY

```

1  C
2  FUNCTION CONE (R,H)
3  C
4  INCLUDE CONST,LIST
5  C
6  C          VOLUME OF CONE (CIRCULAR)
7  CONE = PI*R*R*H / 3.0
8  RETURN
9  C          VOLUME OF CYLINDER (RIGHT=CIRCULAR)
10 ENTRY CYLND (R,H)
11 CONE = PI*R*R*H
12 RETURN
13 C          VOLUME BETWEEN CYLINDER AND SPHEROID
14 C          RROT IS ALONG AXIS OF ROTATION
15 ENTRY CYLSPH (RROT,R)
16 CONE = PI*R*R*RROT / 3.0
17 RETURN
18 C          VOLUME OF FRUSTRUM OF CONE (CIRCULAR)
19 ENTRY FRCONC (R,H,R2)
20 CONE = PI*H*(R*R + R2*R2 + R*R2) / 3.0
21 RETURN
22 C          VOLUME OF HEMISPHERE OR HALF OF SPHEROID
23 C          RROT IS ON AXIS OF ROTATION
24 ENTRY HSPHER (RROT,R)
25 CONE = PI203*R*R*RROT
26 RETURN
27 C          VOLUME OF SPHERE OR SPHEROID
28 C          RROT IS ALONG AXIS OF ROTATION
29 ENTRY SPHERE (RROT,R)
30 CONE = 2.0*PI203*R*R*RROT
31 RETURN
32 C
33 C          AREA OF CYLINDER
34 ENTRY ARACYL (R,H)
35 CONE = 2.0*PI*R*H
36 RETURN
37 C
38 C          AREA OF FRUSTRUM
39 ENTRY AREAFR (R,H,R2)
40 CONE = PI*(R+R2)*SQRT (H*H+(R-R2)**2)
41 RETURN
42 C
43 C          AREA OF HALF OF SPHEROID
44 C          RROT ALONG AXIS OF ROTATION
45 ENTRY ARSPHR (RROT,R)
46 IF (RROT .LE. R) GO TO 110
47 C          ROTATED ABOUT MAJOR AXIS
48 C          E = ECCENTRICITY FOR ELLIPSE
49 E = SQRT (RROT*RROT - R*R)/RROT
50 CONE = PI*R*(R+RROT*ASIN(E)/E)
51 RETURN
52 110 IF (RROT .EQ. R) GO TO 120
53 C          ROTATED ABOUT MINOR AXIS
54 E = SQRT (R*R - RROT*RROT) / R
55 CONE = PI*(R+R*(RROT*RROT/(2.*E))*ALOG((1.+E)/(1.-E)))
56 RETURN
57 C          AREA OF HEMISPHERE

```

***** GOMTRY *****
58 120 CONE = 2.4PI*RR
59 RETURN
60 END

SUBROUTINE HEATEX

```

1      C
2      C
3      C      * * * * *
4      C      * HEAT EXCHANGER DESIGN ROUTINE *
5      C      * UNIVAC 1108      FORTRAN V *
6      C      * DATE CODED      JULY 27, 1971 *
7      C      * J. MCKAY      BLDG 201  X45178 *
8      C      * REVISION 1      JUNE 1972 *
9      C      * REVISION 2      AUGUST 1972 *
10     C      * * * * *
11     C
12     C      SUBROUTINE HEATEX(IGAS,JHEX,WDOTCI,THIN,TCIN,THOUT,TCOUT,PHIN,
13     C      PCIN,PHOUT,PCOUT,OFR,WDOTHI,TOTHHX)
14     C
15     C      REAL ICIN,ICOUT,NTU
16     C
17     C      INCLUDE CIOUNT
18     C      INCLUDE CHEX
19     C      INCLUDE TABLOK
20     C
21     C      DIMENSION F1(2),F2(2),F3(2),F4(2),F5(2)
22     C
23     C      DATA F1/700.,180./ F2/160.,35./ F3/278.2,59.4/ F4/791.4,187.5/
24     C      F5/500.,350./
25     C
26     C      6010 FORMAT (10'T10,1'***** SPECIFY COLD FLUID TYPE *****1)
27     C      6020 FORMAT (10'T10,1'***** ERROR TERMINATION IN HEATEX *****1)
28     C      6030 FORMAT (10'T10,1'*** T=HOT(OUT) > T=HOT(IN) ***1)
29     C      6040 FORMAT (10'T10,1'*** T=HOT(IN) MUST BE < 3500 DEG-R ***1)
30     C      6050 FORMAT (10'T10,1'*** TCR =1F7.3,1 OUT OF RANGE -- HEX. NO.1I3,
31     C      1' GAS TYPE1I2,1 SUBUNIT1I2,1 ***1)
32     C      6060 FORMAT (10'T10,1'*** PCIN =1F9.2,1 NOT IN RANGE ***1)
33     C      6070 FORMAT (10'T10,1'*** TCIN =1F9.2,1 < 1F9.2,1 ***1)
34     C      6100 FORMAT (10'T10,1'*** TCOUT =1F9.2,1 < 1F9.2,1 ***1)
35     C
36     C
37     C      DO 3000 JKM=1,3
38     C      NCVG = 0
39     C
40     C      INITIALIZE OUTPUT VARIABLES
41     C
42     C      DO 12 I = 1,3
43     C      TCRU (JHEX,I,IGAS) = 0.0
44     C      FWDTHU(JHEX,I,IGAS) = 0.0
45     C      FDPDU (JHEX,I,IGAS) = 0.0
46     C      CRU (JHEX,I,IGAS) = 0.0
47     C      NTUU (JHEX,I,IGAS) = 0.0
48     C      UAU (JHEX,I,IGAS) = 0.0
49     C      WOUAU (JHEX,I,IGAS) = 0.0
50     C      WTHXU (JHEX,I,IGAS) = 0.0
51     C      12 CONTINUE
52     C
53     C      *** THE O/F RATIO IS RESTRICTED TO ONE, PCOUT AND PHOUT ***
54     C      *** MUST BE SPECIFIED ***
55     C
56     C      OF = 1.
57     C      WDOTC = WDOTCI

```

```

***** HEATEX *****
58      WDOTHI = 0.
59      C      IGAS = 1 OXYGEN , = 2 HYDROGEN
60      IF (IGAS .GT. 0 .AND. IGAS .LT. 9) GO TO 110
61      WRITE (107,6010)
62      RETURN
63      110 CONTINUE
64      C      CALC. PRESSURE DROPS
65      DPH = PHIN - PHOUT
66      DPC = PCIN - PCOUT
67      C      CHECK FOR ACCEPTABLE PRESSURE DROPS
68      IF (DPH .LE. 0.4*PHIN) GO TO 120
69      DPH = 0.4*PHIN
70      PHOUT = PHIN - DPH
71      120 IF (DPC .LE. 0.25*PCOUT) GO TO 130
72      DPC = 0.25 * PCOUT
73      PCIN = PCOUT + DPC
74      130 CONTINUE
75      C      CALC. TEMPERATURE DROPS
76      DTC = TCOUT - TCIN
77      DTH = THIN - THOUT
78      DHCIN = THIN - TCIN
79      C
80      FWDP = 1.0
81      FWOE = 1.0
82      C      CALCULATE ENTHALPY OF COLD FLUID
83      CALL ENTHOH (PCIN,TCIN,IGAS,ICIN)
84      CALL ENTHOH (PCOUT,TCOUT,IGAS,ICOUT)
85      C      FIND CHANGE IN ENTHALPY (COLD FLUID)
86      DIC = ICOUT - ICIN
87      C      CALC. P(SAT)
88      PSAT = .126 * PHOUT * OF
89      C      LOOKUP T(SAT)
90      CALL FINTAB (NTRID(23))
91      THSAT = MIPE (1,PSAT)
92      C      CHECK FOR POSSIBLE CONDENSATION
93      IF (THOUT .GT. THSAT+30.) GO TO 230
94      THOUT = THSAT + 30.
95      DTH = THIN - THOUT
96      230 IF (THOUT .LT. THIN) GO TO 240
97      WRITE (107,6040)
98      GO TO 4000
99      240 IF (THIN .LE. 3500.) GO TO 250
100     WRITE (107,6050)
101     GO TO 4000
102     250 CONTINUE
103     C      CALCULATE EFFECTIVENESS
104     ISW = 1
105     TCOSAV = TCOUT
106     EFFC = DTC / DHCIN
107     260 EFFH = DTH / DHCIN
108     C      CALC. EFFECTIVENESS SUM
109     EFSUM = EFFC + EFFH
110     C
111     C      CHECK RANGE OF EFFECT. SUM
112     IF (EFSUM .GE. 0.9) GO TO 300
113     GO TO (270,280),ISW
114     270 THOUT = THSAT + 30.
115     DTH = THIN - THOUT

```

```

***** HEATEX *****
116      ISW = 2
117      GO TO 260
118      C      IF EFSUM CAN'T BE RAISED BY THOUT, THEN CHANGE THE
119      C      PARTIAL TCOU BY USING A BYPASS
120      280 EFSUM = 0.5
121      XTCO = THOUT - DHCIN/2.0
122      C      THERMAL BYPASS
123      WDOTC = WDOTC*DTC / (XTCO-TCIN)
124      TCOU = XTCO
125      DTC = TCOU - TCIN
126      WRITE (IOT,6030)
127      CALL ENTHOH (PCOUT,TCOUT,IGAS,ICOUT)
128      DIC = ICOUT - ICIN
129      300 IF (EFSUM .LE. 0.91) GO TO 310
130      EFSUM = 0.91
131      THOUT = DTC + THIN - EFSUM*DHCIN
132      DTH = THIN - THOUT
133      310 CONTINUE
134      C      OF INPUT -- LOOKUP C SUB P (COMB.PROD.)
135      ISW2 = 1
136      CALL FINTAB (NTBID(24))
137      312 XTAB(1) = (THOUT + THIN) / 2.0
138      XTAB(2) = OF
139      ACSBPH = MIPE (2,XTAB)
140      DIH = ACSBPH*DTH
141      WDOTH1 = WDOTC*DIC/DIH
142      ACSBPC = DIC / DTC
143      CC = WDOTC*ACSBPC
144      CH = WDOTH1*ACSBPH
145      CMIN = CC
146      IF (CHIN .GT. CH) CMIN = CH
147      C      FIND EFFECTIVENESS
148      EFFT = CC*DTC / (CMIN*DHCIN)
149      IF (EFFT .LT. 0.9) GO TO 320
150      GO TO (314,314,320),ISW2
151      314 ISW2 = ISW2 + 1
152      EFFT = 0.895
153      THOUT = THIN - EFFT*CMIN / CH*DHCIN
154      DTH = THIN - THOUT
155      GO TO 312
156      320 TCSAT = TSAT (PCOUT,IGAS)
157      C
158      C      *** SUBDIVIDE HEAT EXCH. ACCORDING TO HEAT TRANSFER RANGE ***
159      C
160      IF (PCIN .GE. 10.0 .AND. PCIN .LE. 2250.) GO TO 340
161      WRITE (IOT,6080) PCIN
162      GO TO 4000
163      340 IF (PCIN .GT. F1(IGAS)) GO TO 370
164      IF (TCIN .LT. F2(IGAS) .OR. TCIN .GT. TCSAT) GO TO 400
165      IF (ABS(TCOU-TCSAT) .GT. 0.005*TCOUT) GO TO 390
166      C      HAVE BOILING SUB-UNIT ONLY
167      ISU = 8
168      GO TO 460
169      350 IF (TCOUT .LE. 500.) GO TO 360
170      C      HAVE BOILING, PARALLEL, COUNTER-FLOW SUBUNITS
171      ISU = 11
172      GO TO 460
173      C      HAVE BOILING AND PARALLEL-FLOW SUB-UNITS

```

```

***** HEATEX *****
174 360 ISU = 10
175 GO TO 460
176 370 IF (TCIN .LT. F2(IGAS)) OR (TCIN .GE. F3(IGAS)) GO TO 400
177 IF (TCOUT .GT. F3(IGAS)) GO TO 380
178 C HAVE SUPERCRITICAL SUB-UNIT ONLY
179 ISU = 4
180 GO TO 460
181 380 IF (TCOUT .LE. 500.) GO TO 390
182 C HAVE SUPERCRITICAL, PARALLEL, COUNTER-FLOW SUB-UNITS
183 ISU = 7
184 GO TO 460
185 C HAVE SUPERCRITICAL AND PARALLEL FLOW SUBUNITS
186 390 ISU = 6
187 GO TO 460
188 400 IF (TCIN .LT. 500.) GO TO 410
189 C HAVE COUNTER FLOW SUBUNIT ONLY
190 ISU = 1
191 GO TO 460
192 410 IF (PCIN .GE. F4(IGAS)) GO TO 420
193 IF (TCIN .GE. TCSAT) GO TO 430
194 WRITE (IOT,6090) TCIN,TCSAT
195 GO TO 4000
196 420 IF (TCIN .GE. F3(IGAS)) GO TO 430
197 WRITE (IOT,6090) TCIN,F3(IGAS)
198 GO TO 4000
199 430 IF (TCOUT .LE. 500.) GO TO 440
200 C HAVE PARALLEL AND COUNTER FLOW SUBUNITS
201 ISU = 3
202 GO TO 460
203 440 IF (TCOUT .GT. F3(IGAS)) GO TO 450
204 WRITE (IOT,6100) TCOUT,F3(IGAS)
205 GO TO 4000
206 C HAVE PARALLEL FLOW SUBUNIT ONLY
207 450 ISU = 2
208 C HEAT EXCHANGER NOW SUB-DIVIDED
209 460 CONTINUE
210 C
211 NSU = 0
212 DO 470 I=4,1,-1
213 NKU = 2**(I-1)
214 IF (AND(ISU,NKU) .NE. NKU) GO TO 470
215 NSU = NSU + 1
216 NSS = 1
217 470 CONTINUE
218 IF (NSS .EQ. 4) NSS = 3
219 NST = 4 - NSS
220 NSS = NST + 1 - NSU
221 C
222 C *** NOW GO THROUGH REQUIRED SUBUNITS AND COMPUTE PARAMETERS ***
223 C
224 FPCIN = PCIN
225 FTCIN = TCIN
226 SDPC = 0.
227 TOTWHX = 0.
228 C
229 DO 2000 IS=NSS,NST
230 C THREE POSSIBLE SUBUNITS
231 C SET TEMP., AND PRES. LIMITS EACH UNIT

```

HEATEX

```

232      GO TO (480,530,570),IS
233      C      BOILING OR SUPERCRITICAL
234      480 IF (ISU .LE. 7) GO TO 490
235      C      BOILING
236      FTCOUT = TSAT (PCIN-DPC/2.,IGAS)
237      GO TO 590
238      C      SUPERCRITICAL
239      490 FTCOUT = TCOUT
240      IF (TCOUT .GT. F3(IGAS)) FTCOUT = F3(IGAS)
241      GO TO 590
242      C      PARALLEL FLOW UNIT
243      530 IF (ISU .LE. 3) GO TO 540
244      FTCIN = FTCOUT
245      FPCIN = FPCOUT
246      540 FTCOUT = TCOUT
247      IF (AND(ISU,1) .NE. 0) FTCOUT = 500.
248      GO TO 590
249      C      COUNTER FLOW UNIT
250      570 IF (ISU .EQ. 1) GO TO 580
251      FTCIN = FTCOUT
252      FPCIN = FPCOUT
253      580 FTCOUT = TCOUT
254      590 CALL ENTHOH (FPCIN,FTCIN,IGAS,FIIN)
255      IF (DPC .GT. 0.) GO TO 600
256      FDPIC = 0.2*FPCIN / NSU
257      GO TO 610
258      600 CALL ENTHOH (FPCIN-DPC/2.,FTCOUT,IGAS,FIOT)
259      FDPIC = DPC / DIC * (FIOT-FIIN)
260      610 CALL ENTHOH (FPCIN-FDPC,FTCOUT,IGAS,FIOT)
261      FDIC = FIOT-FIIN
262      FDPIC = DPC / DIC * FDIC
263      IF (IS .EQ. 1) FDPIC = FDPIC/2.
264      FPCOUT = FPCIN - FDPIC
265      FWDTH = WDOTC*FDIC/DIH
266      C      PRES = TEMP SUBUNIT RANGES HAVE BEEN SET
267      IERR = 0
268      IF (IS .NE. 3) GO TO 630
269      C      COUNTER FLOW
270      CALL TCRCL (THIN,THOUT,FTCOUT,FTCIN,TCRMIN,TCRMAX,IERR)
271      GO TO 640
272      C      SUPERCRITICAL, BOILING OR PARALLEL
273      630 CALL TCRCL (THIN,THOUT,FTCIN,FTCOUT,TCRMIN,TCRMAX,IERR)
274      640 IF (IERR .NE. 0) GO TO 4000
275      C      DETERMINE NUMBER OF HEAT TRANSFER UNITS
276      FDTIC = FTCOUT - FTCIN
277      FEMX = DTH
278      IF (FDTIC .GT. DTH) FEMX = FDTIC
279      FEMX = FEMX/(THIN-FTCIN)
280      FDIH = WDOTC*FDIC/FWDTH
281      AVCPC = FDIC/FDTIC
282      AVCPH = FDIH/DTH
283      CMIN = WDOTC*AVCPC*3600.
284      CMAX = FWDTH*AVCPH*3600.
285      CR = CMIN
286      IF (CMAX .GE. CMIN) GO TO 670
287      CR = CMAX
288      CMAX = CMIN
289      CMIN = CR

```



```

***** HEATEX *****
290 CR = CR/CHAX
291 GO TO (680,690,700),IS
292 680 IF (ISU .LE. 7) GO TO 690
293 C BOILING
294 NTU = -ALOG (1.0-FEMX)
295 GO TO 710
296 C SUPERCRITICAL OR PARALLEL
297 690 NTU = -ALOG (1.-FEMX-CR*FEMX) / (1.0+CR)
298 GO TO 710
299 C COUNTER FLOW
300 700 NTU = ALOG ((1.-FEMX)/(1.-CR*FEMX)) / (CR-1.)
301 C CALC. UA FOR EACH UNIT
302 710 UA = NTU*CHIN
303 IF (IS .GT. 1) GO TO 720
304 IF (ISU .GT. 7) GO TO 720
305 C CALCULATION FOR SUPERCRITICAL ONLY
306 GDPC = FDPC*F5(IGAS) / FTCOUT
307 CALL TCRCLC (IGAS,GDPC,FPCIN,DPH,PHIN,TCR)
308 IF (TCR .GT. 10.0) TCR = 10.0
309 CALL WOUACL (WOUA)
310 C SPECIAL FORM OF HEIGHT CALC. FOR SUPERCRITICAL
311 WTHX = WOUA*UA*(1.+TCR)/(1.+TCRMAX)*TCRMAX/TCR
312 GO TO 740
313 C
314 C CALC. TCR FOR O2 OR H2
315 720 CALL TCRCLC (IGAS,FDPC,FPCIN,DPH,PHIN,TCR)
316 C IS TCR WITHIN ACCEPTABLE RANGE
317 721 IF (TCR .LE. TCRMAX) GO TO 724
318 C MUST LOWER TCR
319 C LOWER (DPC/PCIN)
320 CALL TCRLOW (FPCOUT,IGO)
321 GO TO (721,722),IGO
322 C TRY RAISING (DPH/PHIN)
323 722 DPH = 2.*DPH
324 IF (DPH .LE. .4*PHIN) GO TO 723
325 PHIN = PHOUT / 0.6
326 DPH = 0.4*PHIN
327 CALL TCRCLC (IGAS,FDPC,FPCIN,DPH,PHIN,TCR)
328 IF (TCR .LE. TCRMAX) GO TO 730
329 NCVG = IS
330 ETCR = TCR
331 GO TO 730
332 723 PHIN = PHOUT + DPH
333 CALL TCRCLC (IGAS,FDPC,FPCIN,DPH,PHIN,TCR)
334 IF (TCR .GT. TCRMAX) GO TO 722
335 C CHECK FOR TCR TOO LOW
336 724 IF (TCR .GE. TCRMIN) GO TO 730
337 C RAISE (DPC/PCIN)
338 CALL TCRRAZ (FPCOUT,IGO)
339 GO TO (724,725),IGO
340 725 CONTINUE
341 NCVG = IS
342 ETCR = TCR
343 C CALC. (W/UA)
344 730 CALL WOUACL (WOUA)
345 C CALCULATE WEIGHT FOR THIS SUB-UNIT
346 WTHX = WOUA * UA
347 C TOTAL HEAT EXCHANGER WEIGHT

```

```

***** HEATEX *****
348 740 TOTWHX = TOTWHX + WTHX
349 C
350 SDPC = SDPC + FDPC
351 C
352 TCRU (JHEX,IS,IGAS) = TCR
353 FWDTHU(JHEX,IS,IGAS) = FWDTH
354 FDPCU (JHEX,IS,IGAS) = FDPC
355 CRU (JHEX,IS,IGAS) = CR
356 NTUU (JHEX,IS,IGAS) = NTU
357 UAU (JHEX,IS,IGAS) = UA
358 WOUAU (JHEX,IS,IGAS) = WOUA
359 WTHXU (JHEX,IS,IGAS) = WTHX
360 C
361 2000 CONTINUE
362 PCIN = PCOUT + SDPC
363 TCOU = TCOU + TCOU
364 IF (NCVG .EQ. 0) GO TO 4010
365 3000 CONTINUE
366 WRITE (IOT,6060) ETCR,JHEX,IGAS,NCVG
367 GO TO 4010
368 4000 CONTINUE
369 WRITE (IOT,6020)
370 CALL EXIT
371 C
372 4010 CONTINUE
373 CPCLDF(JHEX,IGAS) = ACSBPC
374 CPHOTF(JHEX,IGAS) = ACSBPH
375 EPSLNC(JHEX,IGAS) = EFFC
376 EPSLNH(JHEX,IGAS) = EFFH
377 EPSLNS(JHEX,IGAS) = EFSUM
378 IISU(JHEX,IGAS) = ISU
379 NSSK(JHEX,IGAS) = NSS
380 NSUK(JHEX,IGAS) = NSU
381 IF (TOTWHX .LT. 5.0) TOTWHX = 5.0
382 C
383 RETURN
384 C
385 END

```

SUBROUTINE HEXELC

```

1      C
2      C *****
3      C
4      C THE SUBROUTINE COMPUTES THE WEIGHT AND DELTAP FOR ELECTRICALLY
5      C HEATED HEAT EXCHANGERS EMPLOYING THE PROCEDURES SET FORTH IN
6      C AR-DES,REF,MAN,-EXT,PRESSN,SYS,FOR CRYO,STOR,SYSTEMS,AR-71-7535,
7      C SEPT,10,1971, NAS9-10453 (NASA-HOUSTON).
8      C
9      C **** NOTE THAT PROGRAM ASSUMES THE HEATER FLUX (HF=(Q/A)REF.)
10     C * TO BE RATED AT A REFERENCE TEMPERATURE OF 360 DEGREES
11     C * RANKINE. (SEE AR-71-7537,SEC,5) IF A DIFFERENT REFERENCE
12     C * TEMPERATURE IS TO BE USED, THEN 'TREF' MUST BE CHANGED.
13     C **** A NI-FE ALLOY RESISTANCE HEATER ELEMENT IS ASSUMED.
14     C
15     C PROGRAMMED BY - R.F. HAUSMAN, DEPT 62-13, 104, LMSC, 2-19-73
16     C
17     C *****
18     C
19     C SUBROUTINE HEXELC(NGAS,TIN,TOUT,PIN,HF,LDIA,WDOT,RHOGAS,IFIN,
20     C 1      HEXWGT, DELTAP, UOA, DH, HLENGTH)
21     C
22     C REAL MASVEL, LDIA
23     C
24     C INCLUDE CIOUNT
25     C INCLUDE TABLOK
26     C
27     C DH = LDIA * 2.0
28     C T1 = TIN
29     C T2 = TOUT
30     C PI = 3.141593
31     C TREF = 360.0
32     C
33     C PC1 = 736.9      B OXYGEN      CRIT,PRES,
34     C PC2 = 187.5     B HYDROGEN    CRIT,PRES,
35     C PC3 = 492.2     B NITROGEN    CRIT,PRES.
36     C
37     C *** CALCULATE THE MASS VELOCITY
38     C
39     C 10 CONTINUE
40     C
41     C MASVEL = WDOT/(PI*DH)
42     C
43     C *** CALCULATE THE OVERALL HEAT TRANSER COEFFICIENT -U-
44     C
45     C IF(NGAS,EQ,1) KGS = 2
46     C IF(NGAS,EQ,2) KGS = 1
47     C IF(NGAS,EQ,18) KGS = 2
48     C CALL FINTAB (HTBID(41)+KGS)
49     C XTAB(1) = PIN
50     C XTAB(2) = MASVEL
51     C UOA = HIPE(2,XTAB)
52     C
53     C *** CALCULATE THE VARIABLE VALUES FOR HEX LENGTH EQUATION
54     C
55     C TMEAN = (T2 + T1)/2.0
56     C CALL CSUBP(TMEAN,PIN,NGAS,CPBAR)
57     C BONE = PI * DH * HF * TREF

```

```

***** HEXELC *****
58      BPID = 2.0 * (BONE/(PI*DH))
59      PHIONE = (((T2**2)-(T1**2))/BPID)
60      PHITWO = (ALOG(T2/T1))/UQA
61      C
62      C      *** COMPUTE CORRECTION FACTOR BETA FOR PHITWO
63      C
64      IF(NGAS.EQ.1) POPC = PIN/PC1
65      IF(NGAS.EQ.2) POPC = PIN/PC2
66      IF(NGAS.EQ.18) POPC = PIN/PC3
67      C
68      CALL FINTAB (NTBID(44))
69      XTAB(1) = POPC
70      BETA = MIPE(1,XTAB)
71      C
72      C      *** EVALUATE HEX-LENGTH EQUATION
73      C
74      HLENGTH = MASVEL * CPBAR * (PHIONE + BETA*PHITWO)
75      C
76      C
77      20 CONTINUE
78      C
79      C      *** COMPUTE HEX WEIGHT
80      C
81      IF(NGAS.EQ.1) GO TO 30
82      IF(NGAS.EQ.2) GO TO 40
83      IF(NGAS.EQ.18) GO TO 40
84      30 HEXWGT = 0.1519984 * (DH**1.05379) * HLENGTH
85      GO TO 50
86      40 HEXWGT = 0.0950445 * (DH**1.061) * HLENGTH
87      50 CONTINUE
88      C
89      C      *** IF ANTI-BURNOUT FINS ARE USED - COMPUTE FIN WEIGHT
90      C
91      IF(IFIN.EQ.0) GO TO 60
92      C
93      FINWGT = 0.20688721 * (DH**3.19204) * HLENGTH
94      C
95      C      *** COMPUTE HEX TOTAL WEIGHT
96      C
97      HEXWGT = HEXWGT + FINWGT
98      C
99      60 CONTINUE
100     C
101     C      *** COMPUTE HEX DELTAP FOR FLUID CONSIDERED
102     C
103     CALL FINTAB (NTBID(45))
104     XTAB(1) = HLENGTH
105     XTAB(2) = MASVEL
106     SIGDLP = MIPE(2,XTAB)
107     C
108     DELTAP = SIGDLP/(RHOGAS/0.0765)
109     C
110     RETURN
111     C
112     END

```

SUBROUTINE HEXF21

```

1      SUBROUTINE HEXF21(IGAS,QREG,PCIN,HXWT)
2
3      C
4      C *** DATA CONTAINED IN THIS SUBROUTINE CAME FROM THE FOLLOWING
5      C * SOURCE, FREON-21 CRYOGENIC HEAT EXCHANGER PARAMETRIC DATA,-
6      C *** AIRESEARCH REPORT NO. 71-7720, DTD. 9-22-71.
7
8      C IF(IGAS.EQ.2) GO TO 70
9
10     C IF(QREG.LE.50000.0.AND.QREG.GE.10000.0.AND.PCIN.GE.600.) GO TO 10
11
12     C IF(QREG.LT.50000.0.AND.QREG.GE.10000.0.AND.PCIN.LT.600.) GO TO 20
13
14     C IF(QREG.LT.10000.0.AND.QREG.GE. 700.0.AND.PCIN.GE.600.) GO TO 30
15
16     C IF(QREG.LT.10000.0.AND.QREG.GE. 700.0.AND.PCIN.LT.600.) GO TO 40
17
18     C IF(QREG.LE.70000.0.AND.QREG.GT.50000.0.AND.PCIN.GE.600.) GO TO 50
19
20     C IF(QREG.LE.70000.0.AND.QREG.GT.50000.0.AND.PCIN.LT.600.) GO TO 60
21
22     C 10 IF(QREG.LE.50000.0.AND.QREG.GE.40000.0) HXWT = 0.80
23     IF(QREG.LT.40000.0.AND.QREG.GE.30000.0) HXWT = 0.65
24     IF(QREG.LT.30000.0.AND.QREG.GE.20000.0) HXWT = 0.45
25     IF(QREG.LT.20000.0.AND.QREG.GE.10000.0) HXWT = 0.25
26     RETURN
27
28     C 20 IF(QREG.LE.50000.0.AND.QREG.GE.40000.0) HXWT = 0.60
29     IF(QREG.LT.40000.0.AND.QREG.GE.30000.0) HXWT = 0.40
30     IF(QREG.LT.30000.0.AND.QREG.GE.20000.0) HXWT = 0.25
31     IF(QREG.LT.20000.0.AND.QREG.GE.10000.0) HXWT = 0.15
32     RETURN
33
34     C 30 IF(QREG.LT.10000.0.AND.QREG.GE.2000.0) HXWT = 0.30
35     IF(QREG.LT. 2000.0.AND.QREG.GE. 700.0) HXWT = 0.10
36     RETURN
37
38     C 40 IF(QREG.LT.10000.0.AND.QREG.GE.2000.0) HXWT = 0.20
39     IF(QREG.LT. 2000.0.AND.QREG.GE. 700.0) HXWT = 0.08
40     RETURN
41
42     C 50 IF(QREG.LE.70000.0.AND.QREG.GT.50000.0) HXWT = 1.0
43     RETURN
44
45     C 60 IF(QREG.LE.70000.0.AND.QREG.GT.50000.0) HXWT = 0.8
46     RETURN
47
48     C 70 IF(QREG.LE.60000.0.AND.QREG.GE.50000.0) HXWT = 0.7
49     IF(QREG.LT.50000.0.AND.QREG.GE.40000.0) HXWT = 0.8
50     IF(QREG.LT.40000.0.AND.QREG.GE.30000.0) HXWT = 0.65
51     IF(QREG.LT.30000.0.AND.QREG.GE.20000.0) HXWT = 0.50
52     IF(QREG.LT.20000.0.AND.QREG.GE.10000.0) HXWT = 0.40
53     IF(QREG.LT.10000.0.AND.QREG.GE. 2000.0) HXWT = 0.25
54     IF(QREG.LT. 2000.0.AND.QREG.GE.500.0) HXWT = 0.15
55
56     C RETURN
57
58     C END

```

FUNCTION HPTCP

1
2
3
4

FUNCTION HPTCP(PRES,TEMP)
HPTCP=PTHEAT(PRES,TEMP,1)
RETURN
END

1291 0001
1292 0002
1293 0003
1294 0004

FUNCTION HPTCV

```
1  FUNCTION HPTCV(PRES,TEMP)
2  HPTCV=PTHEAT(PRES,TEMP,2)
3  RETURN
4  END
```

```
1295 0001
1296 0002
1297 0003
1298 0004
```

FUNCTION HPTGAM

```
1  FUNCTION HPTGAM(PRES,TEMP)
2  HPTGAM := PTHEAT(PRES,TEMP,1)
3  RETURN
4  END
```

```
1299 0001
1300 0002
1301 0003
1302 0004
```

FUNCTION HPW

```
1      C      *****
2      C      * ROUTINE NAME - HEX PROPELANT WEIGHT CALC. *
3      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
4      C      * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
5      C      * DATE CODED   - 4/21/70                      *
6      C      *****
7      C
8      FUNCTION HPH(JX,IGAS)
9      HPW = 0.
10     RETURN
11     END
```

SUBROUTINE HTLEAK

```
1      SUBROUTINE HTLEAK(TENV,TFLO,TFLH,WSVH)
2      C
3      C      *** THIS IS CURRENTLY A DUMMY ROUTINE
4      C
5      C      WSVH = 1.0000
6      C
7      RETURN
8      END
```

FUNCTION HVAP

```

1      FUNCTION HVAP(N,T,Z)
2      DIMENSION HVB(17),R(17)
3      DATA(HVB(I),I=1,17)/91.62,195.9,71.5,195.9,75.0,219.2,81.9,219.2,
4      81.9,223.7,71.5,589.3,178.1,425.8,123.0,208.7,1.:/
5      DATA(R(I),I=1,17)/48.3,766.8,40.6,766.8,42.0,96.3,28.6,96.3,
6      28.6,55.8,40.6,90.7,16.8,36.9,11.9,33.5,386.3/
7      IF(N.EQ.2.OR.N.EQ.4) GO TO 5
8      HVAP=HVB(N)*((1.-T/TBOIL(N))/(1.-TSTART(N)/TBOIL(N)))**.98
9      -R(N)*T*Z/778.3
10     RETURN
11     5 HVAP=-160.05404+36.424565*T-1.41762691*T*T+.026014121*T*T*T-
12     .20992651E-3*T**4+.3154429E-6*T**5
13     RETURN
14     END

```

B-193

LMSC-A991396

***** HYENTH *****

58	26.2+284.7+283.7+284.8+364.3+357.6+353.2+350.8+350.1+350.8+491.8+424	0054	0058
59	3.8+422.9+420.9+418.9+419.1+26.4+109.2+98.33+91.88+88.34+201.9+188.	0055	0059
60	44+177.2+168.7+162.4+265.3+255.1+246.2+238.6+232.5+325.1+317.4+310.	0056	0060
61	54+304.2+299.3+384.9+378.8+373.4+368.5+364.3+448.8+444.5+440.3+436.	0057	0061
62	6431.8+222.2+174.7+126.4+266.9+233.7+201.9+313.5+288.4+265.3+362.4.	0058	0062
63	7342.7+325.1+414.5+398.7+384.9+465.8+457.3+448.8+36.57+635.35.58	0059	0063
64	8+24.27+11.37+46.52+10.41+24.51+59.14+5.035+39.1+73.14+22.05+54.9	0060	0064
65	99+88.32+40.42+72.09+104.7+59.98+90.22+122.80.6+109.3+140.1+102.2/	0061	0065
66	DATAAD/129.1+159.1+85.55+76.17+66.88+57.67+48.54+39.65+76.14	0062	0066
67	1+67.02+57.86+48.74+39.63+30.42+65.46+56.63+47.73+38.82+2	0063	0067
68	29.88+20.81+53.51+45.09+36.51+27.86+19.15+10.41+40.29+32.3	0064	0068
69	38+24.23+15.89+7.47+1.17+25.74+18.56+10.9+2.96+5.14+13.54+9	0065	0069
70	4.88+3.58+3.41+10.88+18.6+26.64+7.34+12.46+18.68+25.57+32.85+40.42	0066	0070
71	5+25.89+29.54+34.83+41.04+47.82+54.09+45.69+47.61+51.79+57.23+63.43	0067	0071
72	6+70.29+66.61+66.57+69.53+74.12+79.68+86.88+34.86+3.87+94.91.61+96	0068	0072
73	7.51+102.2+129.4+124.6+119.8+115.1+110.3+105.5+100.8+96.08+89	0069	0073
74	891.37+119.4+114.9+110.4+105.8+101.2+96.51+91.88+87.21+82.	0070	0074
75	941+107.4+103.4+99.26+94.98+90.6+86.19+81.71+77.23+72.58/	0071	0075
76	DATAAC/-92.77+89.71+86.23+82.46+78.47+74.37+70.19+65.93+61	0072	0076
77	1.48+73.53+72.8+70.77+67.93+64.6+61.5+57.2+53.28+49.11+41.	0073	0077
78	228+49.57+51.87+50.91+48.74+45.9+42.7+39.24+35.44+39.1+18.9	0074	0078
79	3+26.49+30.41+30.39+28.88+26.55+23.74+20.45+46.52+40.72+37.0	0075	0079
80	41+34.66+33.21+32.38+32.04+32.06+32.49+88.56+79.51+72.99+68.32+65.0	0076	0080
81	51+62.69+61.11+60.1+59.63+126.4+117.8+109.2+103.8+98.33+95.1+91.88	0077	0081
82	690.11+88.34+30.39+29.64+28.88+27.72+26.55+25.15+23.74+22.1	0078	0082
83	77+20.45+23.63+23.33+22.72+21.84+20.77+19.57+18.24+16.82+15	0079	0083
84	815.17+16.47+16.59+16.27+15.65+14.8+13.76+12.58+11.27+9.88	0080	0084
85	9+8.89+9.51+9.56+9.23+8.6+7.75+6.72+5.57+4.14+3.85+2.07/	0081	0085
86	DATAAF/-2.56+2.55+2.18+1.55+1.7+1.6+7.68+5.74+4.76+4.38+4.45	0082	0086
87	1.4.85+5.5+6.34+7.34+16.7+13.93+12.37+11.57+11.3+11.45+11.88+12.53	0083	0087
88	213.52+26.21+22.5+20.28+19.02+18.39+18.24+18.43+18.89+19.7+36.36+31	0084	0088
89	3.44+28.52+26.72+25.7+25.22+25.15+25.4+25.89+46.52+40.72+37.01+34.6	0085	0089
90	46+32.21+32.38+32.04+32.06+32.49+95.17+80.15+65.54+53.23+44.02+37.3	0086	0090
91	58+32.55+28.95+26.21+115.7+104.1+92.37+81.05+70.96+62.55+55.86+50.6	0087	0091
92	62+46.52+132.6+123.2+113.6+104.94+83.86+45.79+09.72.87+67.54+147.7	0088	0092
93	7+139.7+131.4+123.3+115.3+107.6+100.5+94.17+88.56+161.6+154.5+147.3	0089	0093
94	8+140.2+133.2+126.3+119.8+113.7+107.5+174.7+168.7+162.6+156.6+150.5	0090	0094
95	9+144.5+138.5+132.4+126.4+177.4+161.6+143.4+121.8+95.2+186.3+171.7/	0091	0095
96	DATAAG/155.4+138.8+115.7+195.1+181.6+166.8+150.5+132.6+204.1+191.4	0092	0096
97	1177.9+163.3+147.7+212.9+201.2+188.7+175.5+161.6+222.2+210.3+198.5	0093	0097
98	2186.6+174.7+30.41+30.63+30.67+30.59+30.39+22.5+23.09+23.44	0094	0098
99	3+23.6+23.63+13.86+14.96+15.7+16.18+16.47+4.37+6.14+7.4+48	0095	0099
100	48.28+8.89+6.14+3.46+1.54+1.16+1.85+17.79+13.93+11.18+9.17+7.68+30.	0096	0100
101	552+25.28+21.52+18.77+16.7+44.02+37.38+32.55+28.95+26.21+26.48+27	0097	0101
102	6.26+27.92+28.45+28.92+29.3+29.62+29.88+30.1+30.28+30.41+72	0098	0102
103	721.02+22.11+23.1+23.75+24.38+24.9+25.36+25.73+26.06+26.32+8	0099	0103
104	8+26.46+14.94+16.45+17.67+18.68+19.53+20.25+20.85+21.37+21	0100	0104
105	9.81+22.19+22.5+8.03+10.16+11.84+13.21+14.34+15.28+16.09/	0101	0105
106	DATAAH/-16.77+17.35+17.85+18.18+18.01+3.05+5.37+7.22+8.72+9.	0102	0106
107	196+11.01+11.9+12.66+13.31+13.86+9.41+5.12+1.87+61+2.6+4.2	0103	0107
108	23+5.58+6.72+7.69+8.52+9.12+20.71+14.43+10.1+6.72+4.1+1.99+26	0104	0108
109	3+1.19+2.42+3.47+4.37+33.45+25.24+19.25+14.8+11.45+8.74+6.54+4.	0105	0109
110	471+3.17+1.87+88+45.95+36.79+29.44+23.8+19.44+16.07+13.29+11.02+9.	0106	0110
111	512+7.51+6.14+56.31+47.94+39.94+33.35+28.09+23.89+20.51+17.76+15.44	0107	0111
112	6+13.49+11.96+66.66+58.21+50.23+43.14+37.17+32.25+28.21+24.89+22.11	0108	0112
113	7+19.78+17.79+74.56+67.32+59.73+52.62+46.26+40.81+36.2+32.35+29.11	0109	0113
114	826.37+24.15+82.45+75.41+68.34+61.5+55.11+49.35+44.34+40.03+36.35+3	0110	0114
115	93.21+30.52+88.81+82.65+76.12+69.66+63.44+57.66+52.4+47.78+43.74/	0111	0115

***** HYENTH *****

116	DATAAI/40.24,37.27,95.17,89.16,83.15,77.23,71.38,65.54,60.61,55.69	0112	0116
117	1,51.39,47.7,44.02,104.9,98.01,90.23,81.37,71.12,59.16,45.59,31.88,	0113	0117
118	220.95,113.3,107.3,100.8,93.63,85.67,76.79,66.95,56.39,45.95,120.4,	0114	0118
119	3115.8,110.2,104.2,97.75,90.73,83.18,75.1,66.66,127.6,123.6,118.7,1	0115	0119
120	413.4,107.8,102.1,95.85,89.29,82.45,134.8,130.4,126.1,121.8,116.4,1	0116	0120
121	511.1,105.8,100.5,95.17,65.16,51.82,3.47,10.85,14.26,16.57,18.	0117	0121
122	637,-19.82,-21.02,-22.04,-22.92,-23.69,-24.38,-24.99,-25.54,-26.04,	0118	0122
123	7-26.48,74.13,67.69,57.36,39.09,5.99,-3.23,-7.63,-10.52,-12.65,-14.	0119	0123
124	837,-15.78,-16.98,-18.1,-18.9,-19.69,-20.39,-21.02,83.09,77.17,70.3,	0120	0124
125	961.77,50.17,31.98,12.48,3.55,-1.09,-4.43,-6.93,-8.9,-10.49,-11.86/	0121	0125
126	DATAAJ/-13.03,-14.04,-14.94,89.04,84.56,79.06,72.84,65.52,56.54,44.	0122	0126
127	1.99,30.6,17.92,9.94,4.89,1.51,-1.24,-3.43,-5.22,-6.72,-8.03,94.98,	0123	0127
128	290.77,86.16,81.07,75.42,69.01,61.59,52.84,42.58,31.64,22.32,15.46,	0124	0128
129	310.57,6.94,4.28,1.95,0.01,99.84,96.27,92.24,87.87,83.15,77.97,72.25	0125	0129
130	4,65.86,58.68,50.67,42.06,33.59,26.31,20.44,15.88,12.29,9.41,104.7,	0126	0130
131	5101.3,97.67,93.82,89.72,85.3,80.54,75.38,69.76,63.63,56.98,49.93,4	0127	0131
132	62.76,35.94,29.96,24.89,20.71,109.1,105.9,102.6,99.07,95.49,91.5,87	0128	0132
133	7.5,82.95,78.39,73.16,67.98,62.13,56.27,50.2,44.14,38.79,33.45,71.,	0129	0133
134	837.3,0.,0.,0.,84.3,57.3,0.,0.,0.,97.6,72.9,43.,0.,0.,110.9,89.8,62	0130	0134
135	9.,0.,0.,124.2,107.4,83.28,48.,0.,137.5,123.5,105.9,81.,0.,150.7/	0131	0135
136	DATAAK/138.8,124.9,108.5,85.9,164.1,153.6,142.,128.9,113.3,177.4,1	0132	0136
137	167.9,158.,147.,134.8,78.8,62.3,45.4,0.,0.,89.8,78.3,61.,41.9,29.1,	0133	0137
138	298.8,90.1,78.9,60.1,27.8,106.7,99.5,91.1,80.6,65.2,114.,107.8,100.	0134	0138
139	38,92.8,83.1,120.8,115.2,109.2,102.5,95.,19700.,19330.,19260.,23000	0135	0139
140	4.,22040.,21810.,27350.,25260.,24730.,21390.,19940.,19580.,27180.,2	0136	0140
141	53580.,22680.,36230.,28590.,26650.,22560.,21170.,20520.,30320.,2664	0137	0141
142	60.,25020.,42210.,35060.,31640.,32960.,28900.,27090.,26000.,25260.,	0138	0142
143	740130.,34330.,31710.,30130.,29050.,49050.,41170.,37530.,35320.,338	0139	0143
144	810.,59610.,49530.,44710.,41750.,39690.,71190.,59240.,53210.,49420.	0140	0144
145	9,46750.,82920.,69950.,62910.,58310.,55020.,27100.,23410.,22100./	0141	0145
146	DATAAL/21520.,31750.,26350.,24440.,23550.,37610.,30000.,27280.,260	0142	0146
147	170.,44900.,34480.,30730.,29050.,53610.,39900.,34870.,32610.,63710.	0143	0147
148	2,46340.,39810.,36850./	0144	0148
149			
150	P=PRES		
151	IF(P.LT.1.0) P=1.0		
152	IF(P.LT.1.0) P=1.0		
153	T=TEMP		
154	IF(T.LT.90.0) GO TO 10	0147	0153
155	IF(T.LT.3000.0) GO TO 5	0148	0154
156	IF(T.GE.6000.0) T=9999.99999	0149	0155
157	IF(P.LT.30.0) GO TO 3	0150	0156
158	IF(P.LT.80.0) GO TO 2	0151	0157
159	IF(P.LT.500.0) GO TO 1	0152	0158
160	N=1	0153	0159
161	N1=25	0154	0160
162	GO TO 33	0155	0161
163	1 N=2	0156	0162
164	N1=26	0157	0163
165	GO TO 33	0158	0164
166	2 N=3	0159	0165
167	N1=27	0160	0166
168	GO TO 33	0161	0167
169	3 IF(P.LT.5.0) GO TO 4	0162	0168
170	N=4	0163	0169
171	N1=29	0164	0170
172	GO TO 33	0165	0171
173	4 N=5	0166	0172
		0167	0173

B-195

LMSC-A991396

***** HYENTH *****

174	NI=28	0168 0174
175	GO TO 33	0169 0175
176	5 IF(T.LT.180.0) GO TO 70	0170 0176
177	IF(T.LT.500.0) GO TO 6	0171 0177
178	N=6	0172 0178
179	GO TO 33	0173 0179
180	6 N=7	0174 0180
181	GO TO 33	0175 0181
182	70 IF(P.LT.1175.68) GO TO 8	0176 0182
183	IF(P.LT.2645.28) GO TO 7	0177 0183
184	N=8	0178 0184
185	GO TO 33	0179 0185
186	7 N=9	0180 0186
187	GO TO 33	0181 0187
188	8 IF(P.LT.587.84) GO TO 9	0182 0188
189	N=10	0183 0189
190	GO TO 33	0184 0190
191	9 N=11	0185 0191
192	GO TO 33	0186 0192
193	10 IF(P.LT.1175.68) GO TO 12	0187 0193
194	IF(P.LT.2645.28) GO TO 11	0188 0194
195	N=12	0189 0195
196	GO TO 30	0190 0196
197	11 N=13	0191 0197
198	GO TO 30	0192 0198
199	12 IF(T.GE.59.4) GO TO 15	0193 0199
200	N=14	0194 0200
201	IF(P.GE.187.6385) GO TO 30	0195 0201
202	DO 13 I=2,20	0196 0202
203	IF(P-PS(I))14,14,13	0197 0203
204	13 CONTINUE	0198 0204
205	I=20	0199 0205
206	14 TM=TS(I-1)+(TS(I)-TS(I-1))*(P-PS(I-1))/(PS(I)-PS(I-1))	0200 0206
207	IF(T.LE.TM) GO TO 30	0201 0207
208	GO TO 23	0202 0208
209	15 IF(P.LT.293.92) GO TO 20	0203 0209
210	IF(P.GE.587.84) GO TO 18	0204 0210
211	IF(T.GE.72.0) GO TO 17	0205 0211
212	IF(P.LT.440.88) GO TO 16	0206 0212
213	N=19	0207 0213
214	GO TO 33	0208 0214
215	16 N=20	0209 0215
216	GO TO 33	0210 0216
217	17 N=17	0211 0217
218	GO TO 33	0212 0218
219	18 IF(T.LT.75.6) GO TO 19	0213 0219
220	N=15	0214 0220
221	GO TO 33	0215 0221
222	19 N=16	0216 0222
223	GO TO 33	0217 0223
224	20 IF(T.GE.72.0) GO TO 22	0218 0224
225	IF(P.LT.176.352) GO TO 23	0219 0225
226	IF(T.LT.65.7) GO TO 21	0220 0226
227	N=21	0221 0227
228	GO TO 33	0222 0228
229	21 N=22	0223 0229
230	GO TO 33	0224 0230
231	22 N=18	0225 0231

***** HYENTH *****

```

232 GO TO 33
233 23 N=23
234 IF(P.GE.117.568.AND.T.LE.63.) N=24
235 GO TO 33
236 30 F=P/587.84
237 I=F
238 IF(I.GT.8) I=8
239 FI=I
240 F=F-FI
241 FP=1.0-F
242 TQ=FP*TL(I+1)+F*TL(I+2)
243 IF(T.GE.TQ) GO TO 33
244 HYENTH=FP*HS(I+1)+F*HS(I+2)
245 RETURN
246 33 IF(T.LE.5000.) NI=N
247 FP=(P-BP(N))/DP(N)
248 IP=FP
249 IF(IP.GT.MX(N)) IP=MX(N)
250 FI=IP
251 F=FP-FI
252 FP=1.0-F
253 FT=(T-BT(NI))/DT(N)
254 IT=FT
255 FI=IT
256 FF=FT-FI
257 FT=1.0-FF
258 I=IT*JP(N)+IP*LOC(NI)
259 J=I+JP(N)
260 HYENTH=FP*FT*H(I)+F*FT*H(I+1)+FP*FF*H(J)+F*FF*H(J+1)
261 RETURN
262 END

```

```

0226 0232
0227 0233
0228 0234
0229 0235
0230 0236
0231 0237
0232 0238
0233 0239
0234 0240
0235 0241
0236 0242
0237 0243
0238 0244
0239 0245
0240 0246
0241 0247
0242 0248
0243 0249
0244 0250
0245 0251
0246 0252
0247 0253
0248 0254
0249 0255
0250 0256
0251 0257
0252 0258
0253 0259
0254 0260
0255 0261
0256 0262

```


***** SUBROUTINE INTAB

```

1      C      * * * * *
2      C      * ROUTINE NAME - TABLE INPUT ROUTINE *
3      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
4      C      * PROGRAMMER - R. BOLLINGER 1943 102 26933 *
5      C      * DATE CODED - 3/4/70 *
6      C      * REVISED - JUNE 1971 .. DEC. 1971 *
7      C      * PROGRAMMER - J. MCKAY 19-43 201 45178 *
8      C      * * * * *
9      C
10     C      SUBROUTINE INTAB
11     C
12     C      INTEGER TITLE, OFT
13     C
14     C      LOGICAL JP, PAGE
15     C
16     C      INCLUDE CIOUNT
17     C      INCLUDE CTAB
18     C      INCLUDE CTABA
19     C
20     C      DIMENSION TITLE(4), ICMT(14), LABV(3,5), LABX(3), LABY(3), II(5)
21     C
22     C      ***** INPUT FORMATS
23     C
24     C      5000 FORMAT (4A6, 2I6, 6X, I6)
25     C      5010 FORMAT (13A6, A2)
26     C      5020 FORMAT (3A6, I7, 5E10, 0)
27     C      5030 FORMAT (6A6, 2E12, 0)
28     C      5040 FORMAT (3I6)
29     C      5050 FORMAT (6E12, 0)
30     C      5060 FORMAT (5I5)
31     C
32     C      ***** OUTPUT FORMATS
33     C
34     C      6000 FORMAT (10I, T43, 1***** E C H O   O F   T A B L E   D A T A *****
35     C      1 // T54, 4A6, T100, 1TABLE NO. 1I4//)
36     C      6010 FORMAT (10 **ERROR** THE NUMBER OF DIMENSION IS WRONG. ND = 1, I5)
37     C      6020 FORMAT (T26, 13A6, A2)
38     C      6030 FORMAT (10 **ERROR** THE NUMBER OF POINT IS WRONG. NP = 1, I5)
39     C      6040 FORMAT (T36, 3A6, 2X, 5F13, 4)
40     C      6050 FORMAT (10 **ERROR** THE NUMBER OF DATA POINTS IS WRONG. NV = 1, I5)
41     C      6060 FORMAT (10 **ERROR** THE TABLE TYPE IS WRONG. TYPE = 1, I5)
42     C      6070 FORMAT (10 **DIAGNOSTIC** THE NUMBER OF INTERPOLATION POINTS IS WRO
43     C      ING. NIP = 1, I5, 1. NIP IS SET EQUAL TO 1, I5)
44     C      6080 FORMAT (T52, 3A6, 2X, F12, 4)
45     C      6090 FORMAT (T52, F12, 4, 5XF12, 4)
46     C      6100 FORMAT (T52, F12, 4)
47     C      6110 FORMAT (T38, F12, 4, 2XF12, 4, 5XF12, 4, 2XF12, 4)
48     C      6120 FORMAT (T38, F12, 4, 2XF12, 4)
49     C      6130 FORMAT (10T53, 1COEFFICIENTS', 6X, 1COEFFICIENTS', 11)
50     C      6140 FORMAT (10T44, 1X', 13X, 1Y', 16X, 1X', 13X, 1Y')
51     C      6150 FORMAT (10 **ERROR** THE ABOVE TABLE NUMBER IS LESS THAN 0 OR GREAT
52     C      ER THAN 501//)
53     C      6160 FORMAT (10 **DIAGNOSTIC** THE ABOVE TABLE HAS ALREADY BEEN INPUT. T
54     C      HIS TABLE SHALL REPLACE THE PREVIOUS TABLE. 1)
55     C      6170 FORMAT (10 **ERROR** THE TOTAL SIZE OF THE TABLES HAS EXCEEDED 7000.
56     C      1 THE REQUIRED SIZE IS 1, I6, 1. RUN TERMINATED. 1)
57     C      6180 FORMAT (10T5X1TOTAL TABLE STORAGE =1I6)

```

```

***** INTAB *****
58 6200 FORMAT (10,56X, 'TABLE INPUT SUMMARY', // 28X, 'TABLE', 6X, 'TITLE OF TABLE',
59 1 12X, 'NUMBER OF', 6X, 'NUMBER OF', 6X, 'NUMBER OF', // 27X, 'NUMBER', '66,
60 2 'DIMENSIONS', 5X, 'SUBTABLES', 9X, 'WORDS', //)
61 6210 FORMAT (28X, '4, 2X, '4A6, 3I1, 5)
62 6220 FORMAT (10, 'T90, 'NO, 'WORDS THIS TABLE', 14)
63 C
64 C ***** INITIALIZE THE ROUTINE
65 C
66 C NPRT = 0 PRINT ALL TABLES (1 TABLE/PAGE)
67 C NPRT = 1 PRINT NO TABLE OUTPUT
68 C NPRT > 1 PRINT ALL TABLES (NO PAGE EJECT)
69 C NPRT2 > 0 AND NPRT = 1 PRINT SUMMARY
70 C
71 INTAP = NTAPEI
72 IOTAP = NTAPEI
73 READ (IIN, 5060) IFT, OFT, NPRT, NPRT2
74 IF (IFT .EQ. 0) GO TO 5
75 IF (IFT .GT. 2) INTAP = IFT
76 READ (INTAP) TLA, NV
77 READ (INTAP) (TABLE(JKM), JKM=1, NV)
78 REWIND INTAP
79 KMURDI = NV
80 GO TO 350
81 C
82 5 KMURDI = 0
83 IF (NPRT .NE. 1) GO TO 2
84 IF (NPRT2 .EQ. 0) GO TO 2
85 IF (PAGE(0)) WRITE (IOT, 6200)
86 2 CONTINUE
87 C ***** START OF TABLE INPUT LOOP
88 C
89 DO 290 I=1, NTBN
90 KMUD = KMURDI
91 C
92 C ***** INPUT TITLE - THE TABLE TITLE.
93 C * ND = THE NUMBER OF DIMENSIONS OF THIS TABLE.
94 C * NC = THE NUMBER OF COMMENT CARDS FOR THIS TABLE
95 C * IP = PLOT OPTION, NOT IN MATH MODEL
96 C ***** (SEE AUXILIARY TABLE PROGRAM)
97 C
98 READ (IIN, 5000) TITLE, ND, NC, NT
99 IF (ND .LE. 0) GO TO 300
100 IF (NPRT .EQ. 1) GO TO 1001
101 IF (PAGE(0)) WRITE (IOT, 6000) TITLE, NT
102 JP = PAGE (5)
103 1001 CONTINUE
104 KMURDI = KMURDI + 1
105 IF (NT .GT. 0 .AND. NT .LE. NTBN) GO TO 4
106 WRITE (6, 6150)
107 CALL EXIT
108 4 IF (TLA(NT) .LE. 0) GO TO 6
109 WRITE (6, 6160)
110 6 TLA(NT) = KMURDI
111 ITABLE(KMURDI) = ND
112 C
113 C ***** TEST INPUT VALUE.
114 C
115 IF (ND .GT. 1 .AND. ND .LT. 7) GO TO 20

```

```

***** INTAB *****
116      WRITE(6,6010) ND
117      CALL EXIT
118      C
119      C      ***** IF THERE ARE ANY COMMENT CARDS FOR THIS PLOT PRINT
120      C      ***** THEM OUT.
121      C
122      20 IF (NC .LE. 0) GO TO 40
123      C
124      DO 30 I2 = 1,NC
125      READ(5,5010) ICHT
126      IF (NPRT .EQ. 1) GO TO 30
127      IF (PAGE(1)) WRITE (107,6000) TITLE,NT
128      WRITE(6,6020) ICHT
129      30 CONTINUE
130      C
131      C      ***** INITIALIZE THE INPUT OF THN ND - 2 INDEPENDENT
132      C      ***** VARIABLES.
133      C
134      40 NDM2 = ND - 2
135      NXYT = 1
136      IF (NDM2.EQ.0) GO TO 90
137      IF (NPRT .EQ. 1) GO TO 50
138      IF (PAGE(NDM2+1)) WRITE (107,6000) TITLE,NT
139      C
140      C      ***** INPUT ND - 2 INDEPENDEND VARIABLES.
141      C
142      50 DO 80 I2 = 1,NDM2
143      READ(5,5020) (LABV(I,I2),I=1,3),NP,(TAB(I+1,I2),I=1,NP)
144      IF (NP.GT.1.AND.NP.LT.6) GO TO 60
145      WRITE(6,6030) NP
146      CALL EXIT
147      C
148      60 NXYT = NXYT*NP
149      IF (NPRT .EQ. 1) GO TO 70
150      WRITE(6,6040) (LABV(I,I2),I=1,3),(TAB(I+1,I2),I=1,NP)
151      70 CONTINUE
152      KMURDI = KMURDI + 1
153      ITABLE(KMURDI) = NP
154      DO 75 I3=1,NP
155      KMURDI = KMURDI + 1
156      C
157      C      *** STORE THE (ND-2) INDEPENDENT VARIABLES IN TABLE ARRAY
158      C
159      75 ITABLE(KMURDI) = TAB(I3+1,I2)
160      80 CONTINUE
161      C
162      C      ***** IF PLOTTING IS DESIRED, INPUT THE PLOT LABELS AND
163      C      ***** THE MIN/MAX VALUES OF THE INDEPENDENT VARIABLE.
164      C
165      90 READ(5,5030) LABX,LABY,XMIN,XMAX
166      C
167      C      ***** START OF THE LOOP TO INPUT THE VALUES FOR EACH
168      C      ***** OF THE TWO DIMENSIONAL TABLES WHICH MAKE UP THE MAIN
169      C      ***** ND DIMENSIONAL TABLES.
170      C
171      II(1) = -1
172      DO 100 I2 = 2,5
173      100 II(I2) = -1

```

```

***** INTAB *****
174 C
175 DO 280 I2 = 1,NXYT
176 C
177 C ***** INPUT THE 2D TABLE HEADER CARD AND VERIFY THE DATA.
178 C * NV = NUMBER OF DATA POINTS IN THE TABLE
179 C * TYPE = TABLE TYPE (SEE CTAB)
180 C ***** NIP = NUMBER OF POINTS TO BE USED FOR INTERPOLATION.
181 C
182 READ(5,5040) NV,TYPE,NIP
183 IF (NPRT .NE. 0) GO TO 1004
184 IF (PAGE(0)) WRITE (10T,6000) TITLE,NT
185 1004 CONTINUE
186 C
187 IF (NV .GT. 0 .AND. NV .LE. NSBZ) GO TO 110
188 WRITE(6,6050) NV
189 CALL EXIT
190 C
191 110 IF (TYPE.EQ.0.OR.TYPE.EQ.1) GO TO 120
192 WRITE(6,6060) TYPE
193 CALL EXIT
194 C
195 120 IF (TYPE.EQ.0) GO TO 130
196 IF (NIP.GT.1.AND.NIP.LE.NV) GO TO 130
197 WRITE(6,6070) NIP,NV
198 NIP = NV
199 C
200 C ***** OUTPUT THE HEADER DATA TO DRUM
201 C
202 130 ITABLE(KMURDI+1) = NV
203 ITABLE(KMURDI + 2) = TYPE
204 ITABLE(KMURDI+3) = NIP
205 KMURDI = KMURDI + 3
206 C
207 C ***** INPUT, ECHO AND OUTPUT TO DRUM THE 2D TABLES.
208 C
209 IF (NPRT .EQ. 1) GO TO 170
210 II(NDM2) = MOD(II(NDM2) + 1,ITAB(1,NDM2))
211 IF (NDM2.EQ.0) GO TO 170
212 IF (NDM2.EQ.1) GO TO 150
213 NDM3 = NDM2 - 1
214 C
215 DO 140 I3 = NDM3,1,-1
216 IF (II(I3+1).GT.0) GO TO 150
217 II(I3) = MOD(II(I3) + 1,ITAB(1,I3))
218 140 CONTINUE
219 C
220 150 DO 160 I3 = 1,NDM2
221 IX = II(I3) + 2
222 WRITE(6,6080) (LABV(I,I3),I=1,3),TAB(IX,I3)
223 160 CONTINUE
224 C
225 170 IF (TYPE.EQ.1) GO TO 210
226 C
227 C ***** COEFFICIENT TABLE INPUT
228 C
229 READ(5,5050) (XTAB(I),I=1,NV)
230 DO 175 I3 = 1,NV
231 KMURDI = KMURDI + 1

```

```

***** INTAB *****
232 TABLE(KMURDI) = XTAB(I3)
233 175 CONTINUE
234 IF (NPRT .EQ. 1) GO TO 280
235 WRITE(6,6130)
236 C
237 IF (NV.GT.50) GO TO 180
238 N1 = 1
239 N2 = NV
240 GO TO 190
241 C
242 180 N2 = NV - 50
243 WRITE(6,6090) (XTAB(I),XTAB(I+50),I=1,N2)
244 N1 = N2 + 1
245 IF (N1 .GT. 50) GO TO 280
246 N2 = 50
247 190 WRITE(6,6100) (XTAB(I),I=N1,N2)
248 C
249 ***** DESCRETE TABLE INPUT
250 C
251 210 READ(5,5050) (XTAB(I),YTAB(I),I=1,NV)
252 DO 215 I3 = 1,NV
253 KMURDI = KMURDI + 1
254 TABLE(KMURDI) = XTAB(I3)
255 TABLE(KMURDI+NV) = YTAB(I3)
256 215 CONTINUE
257 KMURDI = KMURDI + NV
258 IF (NPRT .EQ. 1) GO TO 1005
259 WRITE(6,6140)
260 1005 CONTINUE
261 IF (NV.GT.50) GO TO 220
262 N1 = 1
263 N2 = NV
264 GO TO 230
265 C
266 220 N2 = NV - 50
267 WRITE(6,6110) (XTAB(I),YTAB(I),XTAB(I+50),YTAB(I+50),I=1,N2)
268 N1 = N2 + 1
269 IF (N1 .GT. 50) GO TO 280
270 N2 = 50
271 230 CONTINUE
272 IF (NPRT .EQ. 1) GO TO 280
273 WRITE (6,6120) (XTAB(I),YTAB(I),I=N1,N2)
274 C
275 280 CONTINUE
276 KWRD = KMURDI - KMURDI
277 IF (NPRT .EQ. 1 .AND. NPRT2 .NE. 0) WRITE (10T,6210) NT,TITLE,
278 ND,NXYT,KWRD
279 IF (NPRT .NE. 1) WRITE (10T,6220) KWRD
280 C
281 290 CONTINUE
282 C
283 300 IF (KMURDI .LE. MXWRD) GO TO 310
284 WRITE(6,6170) KMURDI
285 CALL EXIT
286 310 IF (OFT .EQ. 0) GO TO 350
287 IF (OFT .GT. 2) IOTAP = OFT
288 WRITE (10TAP) TLA,NV
289 WRITE (10TAP) (TABLE(JKM),JKM=1,KMURDI)

```

```

***** INTAB *****
290      END FILE IOTAP
291      REWIND IOTAP
292      350 WRITE (IOT,6180) KMURDI
293      RETURN
294      C
295      END

```

***** SUBROUTINE LIQRES

```

1      C      *****
2      C      * ROUTINE NAME = LIQUID RESIDUALS *
3      C      * DETERMINATION ROUTINE *
4      C      * ROUTINE LANG = FORTRAN V UNIVAC 1108 EXEC 2 *
5      C      * PROGRAMMER = R. BOLLINGER 1943 102 26933 *
6      C      * DATE CODED = 5/15/70 *
7      C      * REVISED = JULY 1972 *
8      C      * PROGRAMMER = J. MCKAY D1943 201 45178 *
9      C      *****
10     C
11     C      SUBROUTINE LIQRES
12     C
13     C      INCLUDE CTANK
14     C
15     C      DIMENSION CST(2,3)
16     C
17     C      DATA (CST(I,1),I=1,2) / ,022 ,.0035 /
18     C      DATA (CST(I,2),I=1,2) / ,001 ,.000858 /
19     C      DATA (CST(I,3),I=1,2) / ,00075,.0001142 /
20     C
21     C      ***** THREE TYPES OF ACQUISITION DEVICES ARE AVAILABLE
22     C
23     C      IDX = 1 SURFACE TENSION DEVICE
24     C      IDX = 2 POSITIVE DISPLACEMENT DEVICE
25     C      IDX = 3 DIELECTROPHORETIC DEVICE
26     C
27     C      DO 1000 I=1,2
28     C      IDX = SATYPE(I)
29     C      IF (I) .EQ. 2 .AND. IDX .GT. 1) GO TO 100
30     C      OXY, SURFACE TENSION ONLY
31     C      IF (WTOTP .LE. 100000.) GO TO 100
32     C      WLR = 2200. + .008*(WTOTP - 100000.)
33     C      GO TO 200
34     C      100 WLR(I) = CST(I,IDX)*WTOTP(I)
35     C      200 IF (WLRT(I,1) .GT. WLR(I)) WLR(I) = WLRT(I,1)
36     C      1000 CONTINUE
37     C      RETURN
38     C      END

```

***** SUBROUTINE LOCAT

```

1      C      *****
2      C      * ROUTINE NAME = LOCATE AND INPUT THE TABLE *
3      C      * TO BE LOOKED-UP *
4      C      * ROUTINE LANG = FORTRAN V UNIVAC 1108 EXEC 2*
5      C      * PROGRAMMER   = R. BOLLINGER 1943 102 26933 *
6      C      * DATE CODED   = 3/9/70 *
7      C      *****
8      C
9      C      SUBROUTINE LOCAT (MNT)
10     C
11     C      ***** EXPLANATION OF THE CALLING SEQUENCE
12     C      *
13     C      * NT = THE TABLE NUMBER WITHIN A MASTER TABLE. THAT IS,
14     C      * THE NUMBER OF WORDS TO BE SKIPPED TO FIND THE
15     C      ***** DESIRED TABLE IS 200*(NT-1).
16     C
17     C      LOGICAL DIAG
18     C
19     C      INCLUDE CIOUNT
20     C      INCLUDE CTAB
21     C      INCLUDE CTABA
22     C
23     C      NT = IABS (MNT)
24     C      IF (DIAG(0,6HLOCATE)) WRITE (IOT,6000) NT,MNT,IDX1
25     C      IDX = IDX1
26     C      NTH1 = NT - 1
27     C      IF(NTH1.LE.0) GO TO 20
28     C
29     C      DO 10 I1 = 1,NTH1
30     C      MM = 1
31     C      IF(ITABLE(IDX+1).EQ.1) MM = 2
32     C      10  IDX = IDX + MM*ITABLE(IDX) + 1
33     C
34     C      20  NV = ITABLE(IDX)
35     C      TYPE = ITABLE(IDX+1)
36     C      NIP = ITABLE(IDX+2)
37     C      IDX = IDX + 2
38     C
39     C      DO 30 I1 = 1,NV
40     C      IDX = IDX + 1
41     C      IF (MNT .LT. 0) GO TO 26
42     C      XTAB(I1) = TABLE(IDX)
43     C      IF(TYPE.EQ.1) YTAB(I1) = TABLE(IDX+NV)
44     C      GO TO 30
45     C      SWITCH DEPENDENT AND INDEP. VARIABLES
46     C      26  XTAB(I1) = TABLE(IDX+NV)
47     C      YTAB(I1) = TABLE(IDX)
48     C      30  CONTINUE
49     C      IF (DIAG(1,6HLOCATE)) WRITE (IOT,6000) NV,TYPE,NIP,IDX,XTAB(I1),
50     C      XTAB(NV),YTAB(1),YTAB(NV)
51     C      RETURN
52     C      6000 FORMAT (1+14X,4I5,6E15.6)
53     C      END

```


***** SUBROUTINE LPROP

```

1      SUBROUTINE LPROP(T,P,D,K,H,S,U,Z)
2      COMMON /RFR/ RF(10)
3      COMMON /METH/ M
4      C
5      C.... ROUTINE TO CALCULATE THE PROPERTIES OF THE LIQUID
6      C
7      C      K=1 INPUT IS T + D
8      C      K=2 INPUT IS T + P
9      C      K=3 INPUT IS T, P + D
10     C
11     IF(M.EQ.2)GO TO 1
12     CALL VPROP(T,P,D,K,H,S,U,Z)
13     RETURN
14     I R=RF(5)
15     AK=RF(6)
16     IF(K.EQ.1)CALL PFND(T,D,P)
17     IF(K.EQ.2)CALL DFND(T,P,D,Z1,0)
18     VP=VPN(T)
19     CALL DFND(T,VP,DSV,Z1,2)
20     CALL VPROP(T,VPP,DSV,1,HSV,SSV,USV,ZSV)
21     VSV = 1.0/DSV
22     CALL DFND(T,VP,DSL,Z1,1)
23     VSL = 1.0/DSL
24     CALL DPDTVP(T,VP,DPDT)
25     HSL=HSV-T*DPDT*(VSV-VSL)*AK
26     SSL=SSV+(HSL-HSV)/T
27     USL=USV+(HSL-HSV)-(VP*(VSL-VSV))*AK
28     DLD = ALOG(D)
29     DLS = ALOG(DSL)
30     FID=R*DLD-FING1(T,D)
31     FIS=R*DLS-FING1(T,DSL)
32     F2D=FING2(T,D)+R*T*DLD
33     F2S=FING2(T,DSL)+R*T*DLS
34     S=SSL-(FID-FIS)*AK
35     U=USL+((F2D-F2S)-T*(FID-FIS))*AK
36     H=U + (P/D)*AK
37     Z = P/(D*R*T)
38     RETURN
39     END

```

***** SUBROUTINE LPROP

```

1      SUBROUTINE LPROP(TB,PB,DB,K,HB,SB,UB,ZB)
2      COMMON /RFPR/RF(10)
3      WT=RF(7)
4      T = TB/1.8
5      P = PB * 6.8947572E+3 / 1.01325E+5
6      D = DB * 453.59237E-3 / (WT * 2.8316847E-2)
7      CALL LPROP(T,P,D,K,H,S,U,Z)
8      PB = P * 1.01325E+5 / 6.8947572E+3
9      DB = D * WT * 2.8316847E-2 / 453.59237E-3
10     HB = H * 453.59237 / (1.0543502E+3 * WT)
11     UB = U * 453.59237 / (1.0543502E+3 * WT)
12     SB = S * 453.59237 / (1.0543502E+3 * 1.8 * WT)
13     ZB = Z
14     RETURN
15     END

```

***** SUBROUTINE LSSCMP

```

1      SUBROUTINE LSSCMP
2      C
3      C *** THIS SUBROUTINE PROVIDES ECLSS CONFIGURATION COMPONENT
4      C * SIZING AND PRESSURE DROP DATA FOR OXYGEN AND NITROGEN
5      C *** FLUIDS.
6      C
7      C
8      C INTEGER GSTATE
9      C
10     C LOGICAL PAGE, JP
11     C
12     C
13     C INCLUDE CACCUH
14     C INCLUDE CCNFIG
15     C INCLUDE CCNTRL
16     C INCLUDE CDCYCL
17     C INCLUDE CECLSS
18     C INCLUDE CENG
19     C INCLUDE CHEX
20     C INCLUDE CHSORC
21     C INCLUDE CIOUNT
22     C INCLUDE CNAMES
23     C INCLUDE CONST
24     C INCLUDE CTANK
25     C
26     C ***** INITIALIZE THE ROUTINE
27     C
28     C   IDX = 0
29     C   ISIGN = 1
30     C   JKM = 0
31     C   NIENH = 0.0
32     C   WGGTOT(1) = 0.0
33     C   WGGTOT(2) = 0.0
34     C   HFTOT(1) = 0.0
35     C   HFTOT(2) = 0.0
36     C   QTREQ(1) = 0.0
37     C   QTREQ(2) = 0.0
38     C   CI = 1152.0/(GRAVITY*PI**2)
39     C   IF (PAGE(0)) WRITE (IOT,6050)
40     C   WRITE (IOT,6020)
41     C   JP = PAGE (3)
42     C
43     C ***** START OF CONFIGURATION PROCESSING LOOP
44     C
45     C   DO 1000 I=1,ICNF
46     C     IDX = IDX + ISIGN
47     C     MACH(IDX) = 0.0
48     C     MELG(IDX) = 6H
49     C     CALL GETCON(IDX)
50     C
51     C ***** BRANCH TO THE REQUIRED CONFIGURATION TYPE, SEE CCNFIG
52     C ***** FOR BRANCH DEFINITIONS.
53     C
54     C   GO TO (100,200,300,400,450,500,450,450,400,405,600,700,800,900,
55     C     1,230,250,270,1100), CFUNCT
56     C
57     C ***** SETUP THE GAS TYPE *****

```

```

***** LSSCMP *****
58 C
59 100 IGAS = CFTYPE
60 GSTATE = ICNFIG(5)
61 IF (IGAS .EQ. JKM) GO TO 110
62 JKM = IGAS
63 ISIGN = 1
64 ISTRT(IGAS) = IDX + 1
65 JX = 0
66 110 CONTINUE
67 C
68 IF (IGAS.EQ. 2.AND.GSTATE.EQ.1) GO TO 111
69 GO TO 112
70 111 IF (PAGE(0)) WRITE (IOT,6051)
71 WRITE (IOT,6020)
72 JP = PAGE(3)
73 C
74 112 CONTINUE
75 C
76 IF (11.EQ.1) GO TO 999
77 IF (IGAS.EQ. 2.AND.GSTATE.EQ.1) GO TO 999
78 PRES(IDX) = PRES(IDX - ISIGN)
79 WDOTN(IDX) = WDOTN(IDX-ISIGN)
80 TEMP(IDX) = TEMP(IDX - ISIGN)
81 GO TO 999
82 C
83 ***** PROCESS THE ECLSS *****
84 C
85 270 WDOTN(IDX) = WDOTI(IGAS)
86 PRES(IDX) = PLSNOM(IGAS)
87 TEMP(IDX) = TLSNOM(IGAS)
88 GO TO 999
89 C
90 ***** PROCESS A LINE *****
91 C
92 300 FLD = FRCOEF(IDX)*LOD(IDX)/DIAM(IDX)
93 LDV = CFTYPE/10
94 CFTYPE = CFTYPE - LDV * 10
95 310 WDOTN(IDX) = WDOTN(IDX-ISIGN)
96 TEMP(IDX) = TEMP(IDX-ISIGN)
97 GO TO 510
98 C
99 ***** PROCESS A CONTROL *****
100 C
101 400 FLD = FRCOEF(IDX)*LOD(IDX)
102 IDV = CFTYPE /10
103 CFTYPE = CFTYPE - IDV * 10
104 DIAM(IDX) = AMINI(DIAM(IDX+1),DIAM(IDX-1))
105 GO TO 310
106 C
107 ***** PROCESS A REGULATOR *****
108 C
109 405 FLD = FRCOEF(IDX)*LOD(IDX)
110 IDV = CFTYPE /10
111 CFTYPE = CFTYPE - IDV * 10
112 DIAM(IDX) = AMINI(DIAM(IDX+1),DIAM(IDX-1))
113 WDOTN(IDX) = WDOTN(IDX-ISIGN)
114 TEMP(IDX) = TEMP(IDX-ISIGN)
115 C

```

```

***** LSSCMP *****
116      IX = IDX - ISIGN
117      IF (APRES(IGAS) .EQ. 0.0) GO TO 406
118      DLPREG = (APRES(IGAS) - ANDELP(IGAS)/2.0) - PRES(IX)
119      PRES(IX) = PRES(IX) + ISIGN * DLPREG
120      GO TO 561
121      C
122      406 CONTINUE
123      DLPREG = HEXCOP(1,IGAS) - PRES(IX)
124      PRES(IX) = PRES(IX) + ISIGN * DLPREG
125      GO TO 561
126      C
127      C ***** PROCESS A FITTING *****
128      C
129      450 FLD = FRCOEF(IX) * LOD(IX)
130      LDV = CFTYPE/10
131      CFTYPE = CFTYPE - LDV * 10
132      DIAM(IX) = AMINI(DIAM(IX+1),DIAM(IX-1))
133      GO TO 310
134      C
135      C ***** PROCESS A TAP *****
136      C
137      500 WDOTN(IX) = WDOTT(IGAS)
138      LDV = CFTYPE/10
139      CFTYPE = CFTYPE - LDV * 10
140      FLD = FRCOEF(IX)*LOD(IX)
141      TEMP(IX) = TEMP(IX-ISIGN)
142      DIAM(IX) = AMINI(DIAM(IX+1),DIAM(IX-1))
143      C
144      C ***** COMPUTE LINE, CONTROL, FITTING OR TAP DELTA PRESSURE.
145      C
146      510 IX = IDX - ISIGN
147      C
148      C ***** DELTA PRESSURE WHEN GASEOUS
149      C
150      C CALC. RHO OF GAS
151      520 IF (IGAS.EQ.2) GO TO 521
152      CALL DENSON(TEMP(IX),PRES(IX),1,RHO,ZEO)
153      GO TO 522
154      521 CALL DENSON(TEMP(IX),PRES(IX),18,RHO,ZEN)
155      522 CONTINUE
156      C
157      DELP = CT*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IX)**4)
158      C
159      C ***** IF PCT. OF PRESSURE CHANGE EXCEEDS ONE PCT. - RECOMPUTE
160      C ***** DELTA-P, IF NOT, COMPUTE THE NEW PRESSURE
161      C
162      IF (DELP/(PRES(IX) + DELP) - 0.01) 560,560,530
163      C
164      C CALC. RHO OF GAS
165      530 IF (IGAS.EQ.2) GO TO 531
166      CALL DENSON(TEMP(IX),PRES(IX)+DELP/2.0,1,RHO,ZEO)
167      GO TO 532
168      531 CALL DENSON(TEMP(IX),PRES(IX)+DELP/2.0,18,RHO,ZEN)
169      532 CONTINUE
170      C
171      DELP = CT*FLD*(WDOTN(IX)/CNOPER)**2/(RHO*DIAM(IX)**4)
172      C
173      C ***** AGAIN CHECK PCT. OF PRESSURE CHANGE. IF PCT. EXCEEDS

```

```

***** LSSCMP *****
174 C      * 2.8 PCT. COMPUTE THE DELTA-P BY USE OF THE COMPRESSIBLE
175 C      ***** FLOW EQUATIONS. (REF.-RPL-TDR-64-25-VOL.1,REV.D)
176 C
177 C      IF(DELPH/(PRES(IX) + DELP) = 0.028)560,560,540
178 C
179 C      540 A = PI*DIAM(IDX)**2/576.0
180 C      IF(IGAS.EQ.2) GO TO 541
181 C      CALL COMELO(IDX,PRES(IX),TEMP(IX),FLD,A,WDOTN(IX)/CNOPER,1,DELP)
182 C      GO TO 542
183 C      541 CALL COMELO(IDX,PRES(IX),TEMP(IX),FLD,A,WDOTN(IX)/CNOPER,18,DELP)
184 C      542 CONTINUE
185 C
186 C      PRES(IDX) = PRES(IX) + ISIGN * DELP
187 C      GO TO 561
188 C
189 C      ***** COMPUTE NEW PRESSURE
190 C
191 C      560 PRES(IDX) = PRES(IX) + ISIGN*DELP
192 C
193 C      ***** COMPUTE THE GAS MACH NUMBER
194 C
195 C      CALC. RHO OF GAS
196 C      IF(IGAS.EQ.2) GO TO 551
197 C      CALL DENSON(TEMP(IX),PRES(IDX),1,RHO,ZEO)
198 C      GO TO 552
199 C      551 CALL DENSON(TEMP(IX),PRES(IDX),18,RHO,ZEN)
200 C      552 CONTINUE
201 C
202 C      IF(IGAS.EQ.2) GO TO 554
203 C      CALL VGVS(IDX,RHO,1)
204 C      GO TO 555
205 C      554 CALL VGVS(IDX,RHO,18)
206 C      555 CONTINUE
207 C
208 C      561 CONTINUE
209 C
210 C      ***** COMPUTE LINE WEIGHT
211 C
212 C      IF(CFUNCT.EQ.3) GO TO 562
213 C      IF(CFUNCT.EQ.5) GO TO 562
214 C      IF(CFUNCT.EQ.6) GO TO 562
215 C      IF(CFUNCT.EQ.7) GO TO 562
216 C      IF(CFUNCT.EQ.8) GO TO 562
217 C      GO TO 570
218 C
219 C      562 CALL LWEGHT(IDX,LDV)
220 C
221 C      GO TO 999
222 C
223 C      ***** COMPUTE CONTROL, FITTING OR TAP WEIGHT
224 C
225 C      570 HEIGHT(IDX) = CFTH (DIAM(IDX),PRES(IDX),IDV)
226 C      GO TO 999
227 C
228 C      ***** PROCESS AN ACCUMULATOR *****
229 C
230 C      600 PRES(IDX) = APRES(IGAS)
231 C      TEMP(IDX) = TEMP(IDX) - ISIGN)

```

```

***** LSSCMP *****
232      WDOTN(IDX) = WDOTN(IDX - ISIGN)
233      INDXAC(IGAS) = IDX
234      GO TO 999
235      C
236      C      ***** PROCESS A TANK OR SUPPLY *****
237      C
238      700 CFT = 1
239      INDXTK(IGAS) = IDX
240      C
241      IF(SIPRES(IGAS,CFT)) 720,710,720
242      C
243      C      ***** IF NO TANK INPUT PRESSURE IS INPUT USE THE VALUE CALC.
244      C
245      710 SIPRES(IGAS,CFT) = PRES(IDX-ISIGN)
246      GO TO 740
247      C
248      C      ***** CHECK THE TANK INPUT PRESSURE AGAINST THE REQUIRED
249      C      * CALCULATED PRESSURE. IF THE TANK INPUT PRESSURE IS LESS
250      C      * THAN THE CALC. PRESSURE WRITE A DIAGNOSTIC MESSAGE AND
251      C      * SET THE TANK INPUT PRESSURE = THE REQUIRED PRESSURE.
252      C      ***** IF NOT CONTINUE WITH CALCULATIONS.
253      C
254      720 IF(SIPRES(IGAS,CFT) - PRES(IDX-ISIGN)) 730,740,740
255      C
256      730 WRITE (6,6000) SIPRES(IGAS,CFT), PRES(IDX-ISIGN)
257      GO TO 710
258      C
259      740 PRES(IDX) = SIPRES(IGAS,CFT)
260      C
261      C      ***** DO THE SAME CHECKS FOR THE INPUT TANK TEMPERATURE.
262      C
263      IF(SITEMP(IGAS,CFT)) 760,750,760
264      750 SITEMP(IGAS,CFT) = TEMP(IDX-ISIGN)
265      C
266      760 TEMP(IDX) = SITEMP(IGAS,CFT)
267      C
268      WDOTN(IDX) = WDOTN(IDX-ISIGN)
269      HEIGHT(IDX) = WTTOT(IGAS)
270      GO TO 999
271      C
272      800 GO TO 1000
273      C
274      C      ***** PROCESS A HEAT EXCHANGER *****
275      C
276      900 IF(ISIGN.GT.0) GO TO 910
277      C
278      WRITE (107,6005) ISIGN
279      C
280      910 CONTINUE
281      JX = JX + 1
282      JHX = JX
283      WDOTN(IDX) = WDOTN(IDX-ISIGN)
284      WDOTCF(JX,IGAS) = WDOTN(IDX)
285      UCODE(JX,IGAS) = CODE(IDX)
286      C
287      TEMP(IDX) = HEXCIT(JX,IGAS)
288      DLPRES = HXCOLDP(JX,IGAS)
289      PRES(IDX) = PRES(IDX-ISIGN) + DLPRES*ISIGN

```

```

***** LSSCMP *****
290      WEIGHT(IDX) = WHXTOT(JX,IGAS)
291      C
292      C      ***** END OF CONFIGURATION PROCESSING LOOP *****
293      C
294      999 CONTINUE
295      IF (.NOT. PAGE(1)) GO TO 1998
296      C
297      C      *** PAGE HEADER HAS BEEN MOVED TO STATEMENT GROUP 100
298      C
299      1998 CONTINUE
300      KFUNCT = FNAME(CFUNCT)
301      WRITE(IOT,6030)KFUNCT,CODE(IDX),CFATYPE,CNOPER,CNSTBY,ISIGN,IDX,
302      1      IGAS,GSTATE,FRCOEF(IDX),LOD(IDX),DIAM(IDX),
303      2      ITHICK(IDX),PRES(IDX),TEMP(IDX),WDOTH(IDX),
304      3      WEIGHT(IDX),MACH(IDX),MFLG(IDX),
305      IF (PRES(IDX) .GE. 0. .AND. TEMP(IDX) .GE. 0.) GO TO 998
306      WRITE (IOT,6040)
307      CALL EXIT
308      998 CONTINUE
309      C
310      200 CONTINUE
311      230 CONTINUE
312      250 CONTINUE
313      1000 CONTINUE
314      1100 CONTINUE
315      C      USED BY WEIGHT SUMMARY OUTPUT
316      KHEND = IDX - 1
317      KOEND = IHSTT - 2
318      C
319      C
320      RETURN
321      C
322      C      ***** OUTPUT FORMATS
323      C
324      6000 FORMAT(10 *DIAGNOSTIC* TANK INPUT PRESSURE IS LESS THAN THE REQUIRED
325      1ED PRESSURE, TANK PRESSURE SET = REQUIRED PRESSURE,1/15X,TANK INPUT
326      2UT PRESSURE = 1,P7.2,1, REQUIRED PRESSURE = 1,P7.2)
327      C
328      6005 FORMAT(10 **ERROR** ISIGN = 1,I3,1THERE IS A CONFIGURATION ERROR//)
329      C
330      6010 FORMAT(10 *ERROR* A PUMP WAS ENCOUNTERED BUT NO TANK CAN BE FOUND.
331      1 PUMP CONFIGURATION INDEX NUMBER = 1,I3)
332      C
333      6020 FORMAT(10 F CODE FT NO NS IS IDX G GS FCOEF L/D
334      1 DIAM ITHICK PRES TEMP WDOT WEIGHT MACH M
335      2FLAG1//1)
336      C
337      6030 FORMAT(2XA3,2XA6,I3,6I4,F9.6,F12.4,2F8.4,2F9.2,F8.4,F8.4,F10.7,
338      13X,A6)
339      C
340      6040 FORMAT (T44, **** TERMINATE = NEGATIVE TEMP. OR PRES. ****)
341      C
342      6050 FORMAT(/T38, **** SUMMARY OF COMPUTED SYSTEM CONFIGURATION PARAMETE
343      1RS ****)
344      C
345      6051 FORMAT(/T32, **** SUMMARY OF COMPUTED SYSTEM CONFIGURATION PARAMETE
346      1RS = CONTD. ****)
347      C

```


***** LSSCMP *****
348 END

***** SUBROUTINE LWEGHT

```

1      C      *****
2      C      * ROUTINE NAME - LINE WEIGHT COMPUTATION *
3      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
4      C      * PROGRAMMER   - R. BOLLINGER 1949 102 26933 *
5      C      * DATE CODED   - 3/17/70
6      C      * REVISED     - 12-17-71 (RFH)
7      C      *****
8      C
9      C      SUBROUTINE LWEGHT(IDX,LDV)
10     C
11     C      ***** EXPLANATION OF THE CALLING SEQUENCE
12     C
13     C      ***** IDX - INDEX OF THE CONFIGURATION TABLE.
14     C
15     C      LOGICAL DIAG
16     C
17     C
18     C      INCLUDE CCNFIG
19     C      INCLUDE CMATRL
20     C      INCLUDE CONST
21     C      INCLUDE TABLOK
22     C
23     C      IF(CMTYPE,LE,5) GO TO 5
24     C      IF(CMTYPE,EQ,6) GO TO 50
25     C      IF(CMTYPE,EQ,7) GO TO 60
26     C
27     C      5 CONTINUE
28     C
29     C      ***** LOOK UP FTU (S1) FOR LINE
30     C
31     C      IF(DIAG(0,6HLWEGHT)) WRITE (6,6000) RHOL(CMTYPE),RHOI(CMTYPE)
32     C
33     C      CALL FINTAB (NTBID(18)+CMTYPE)
34     C      XTAB(1)=TEMP(IDX)
35     C      S1 = MIPE(1,XTAB)
36     C
37     C      ***** COMPUTE THE THICKNESS FOR FTU
38     C
39     C      THKL = PRES(IDX)*DIAM(IDX)*2.5/(2.0*S1)
40     C
41     C      JKM = 0
42     C      IF (PRES(IDX) .GE. 1000.) JKM = 5
43     C      IF (PRES(IDX) .GE. 3000.) JKM = 10
44     C      IF (THKL .LE. MINTHK(CMTYPE+JKM)) THKL = MINTHK(CMTYPE+JKM)
45     C
46     C      ***** COMPUTE THE WGT/FT FOR THE TUBING OR PIPE MATERIAL
47     C
48     C      WGTFT = PI * DIAM(IDX) * THKL * RHOL(CMTYPE)/144.0
49     C
50     C      GO TO (71,72,73,74,75),LDV
51     C
52     C      ***** COMPUTE EQUIVALENT LENGTH OF FITTING
53     C
54     C      ** FOR A 4-WAY TEE
55     C
56     C      72 FLOD = 4.75 * DIAM(IDX)
57     C      GO TO 80

```

```

***** LWEGHT *****
58 C
59 C      ** FOR A 3-WAY TEE
60 C
61 C      73 FLOD := 3.75 * DIAM(IDX)
62 C      GO TO 80
63 C
64 C      ** FOR A 90 DEG. ELBOW
65 C
66 C      74 FLOD := 2.75 * DIAM(IDX)
67 C      GO TO 80
68 C
69 C      ** FOR A 45 DEG. ELBOW
70 C
71 C      75 FLOD := 1.75 * DIAM(IDX)
72 C      GO TO 80
73 C
74 C      ***** COMPUTE THE WEIGHT OF A FITTING
75 C
76 C      80 WEIGHT(IDX) = WGTFT * (FLOD/12.) * 1.25
77 C
78 C      RETURN
79 C
80 C      ***** COMPUTE THE WEIGHT OF A LINE
81 C
82 C      71 WEIGHT(IDX) = WGTFT * LOD(IDX)/12.
83 C
84 C      ***** COMPUTE THE WEIGHT OF THE INSULATION
85 C
86 C      WI(IDX) = PI * ITHICK(IDX) * LOD(IDX) * RHOI(CITYPE) * (DIAM(IDX)
87 C      + ITHICK(IDX)/2.0)/1728.0
88 C
89 C      IF(DIAG(1,6HLWEGHT)) WRITE(6,6000) THKL,SI,WGTFT
90 C      RETURN
91 C
92 C      50 CONTINUE
93 C
94 C      ***** COMPUTE WEIGHT OF VACUUM JACKETED CRES LINE (321/347)
95 C
96 C      AYE1 := 0.217684
97 C      BEE1 := -6.69016E-03
98 C      WGTFT := 1.0/((AYE1)+(BEE1)*DIAM(IDX))
99 C      WEIGHT(IDX) := WGTFT * LOD(IDX)/12.
100 C
101 C      IF(DIAG(1,6HLWEGHT)) WRITE (6,6000) AYE1,BEE1,DIAM(IDX),WGTFT,
102 C      LOD(IDX)
103 C      RETURN
104 C
105 C      60 CONTINUE
106 C
107 C      ***** COMPUTE WEIGHT OF VACUUM JACKETED ALUMINUM LINE (2219)
108 C
109 C      AYE2 := 0.559277
110 C      BEE2 := -2.00888E-02
111 C      WGTFT := 1.0/((AYE2)+(BEE2)*DIAM(IDX))
112 C      WEIGHT(IDX) := WGTFT * LOD(IDX)/12.
113 C
114 C      IF(DIAG(1,6HLWEGHT)) WRITE(6,6000) AYE2,BEE2,DIAM(IDX),WGTFT,
115 C      LOD(IDX)

```

```

***** .LWEGHT *****
116 C
117 RETURN
118 C
119 6000 FORMAT(141,14X,5E15.8)
120 END

```

***** SUBROUTINE MATHAX

```

1  SUBROUTINE PUMPEF (KUP)
2  C
3  INCLUDE CIOUNT
4  INCLUDE SPUMP.LIST
5  INCLUDE TABLOK
6  C
7  DIMENSION PCRIT(2),CRTMP(2)
8  DIMENSION CT1(2),CT2(2),CT3(2),CT4(2),CT5(2),
9  CH1(2),CH2(2),CH3(2),CH4(2),CH5(2)
10 C
11 DATA PCRIT /731.379,187.506/ CRTMP / 278.237,59.956/
12 DATA CT1 /1.1,3/ CT2 /8.82/ CT3 /48.521/ CT4 / .972,414/
13 1 CT5 /2185.2685/ CH1 /31.212/ CH2 /81.79/ CH3 /19.498/
14 2 CH4 /1.0016,000123/ CH5 /0.00513/
15 C
16 6001 FORMAT (10I20X)*** TCR(MIN) > TCR(MAX) CHANGE DESIGN POINT ***
17 C
18 C
19 DO 100 K=1,KUP
20 NSG = NSG + DNS/XNS
21 C
22 CALC. SPECIFIC SPEED (NSS)
23 C
24 NSS = NSG*XNS
25 C
26 FIND HEAD COEF. PSI = F(NSS)
27 C
28 CALL FINTAB (NTBID(19))
29 PSI = MIPE (1,NSS)
30 C
31 FIND ADIABATIC EFF. NUZ = F(NSS)
32 C
33 CALL FINTAB (NTBID(20))
34 NUZ = MIPE (1,NSS)
35 C
36 CALC. IMPELLER TIP SPEED (U)
37 C
38 U = SQRT (32.2*H/PSI)
39 C
40 CALC. IMPELLER DIAMETER
41 C
42 DI = 229*U /NSG
43 C
44 FIND EFF. QUOTIENT
45 C
46 CALL FINTAB (NTBID(21))
47 EFFQ = MIPE (1,DI)
48 C
49 CALC. PUMP HYDRAULIC EFF
50 C
51 NU(K) = EFFQ*NUZ
52 100 CONTINUE
53 C
54 RETURN
55 C
56 USED BY THE HEATEX ROUTINE
57 C

```

```

***** MATHAX *****
58      ENTRY TCRCL (THA,THB,TCA,TCB,TCRMN,TCRMX,IERR)
59      C      CALCULATE TCR MIN
60      TCRA = (THA-1800.)/(1800.-TCA)
61      TCRB = (THB-1800.)/(1800.-TCB)
62      IF (TCRA .LT. 0.) TCRA = 0.
63      IF (TCRB .LT. 0.) TCRB = 0.
64      TCRMN = TCRA
65      IF (TCRA .LT. TCRB) TCRMN = TCRB
66      C      CALCULATE TCR MAX.
67      TCRA = (THA-550.)/(550.-TCA)
68      TCRB = (THB-550.)/(550.-TCB)
69      IF (TCRA .LT. 0.) TCRA = 5000.
70      IF (TCRB .LT. 0.) TCRB = 5000.
71      TCRMX = TCRA
72      IF (TCRA .GT. TCRB) TCRMX = TCRB
73      IF (TCRMX .EQ. 5000.) IERR = 1
74      IF (TCRMN .LT. TCRMX) GO TO 200
75      WRITE (IOT,6001)
76      IERR = 1
77      200 RETURN
78      C
79      C      USED BY THE HEATEX ROUTINE
80      C
81      ENTRY ENTHOH (PE,TE,KG,ENTH)
82      IF (KG .GT. 1) GO TO 300
83      ENTH = OXENTH (PE,TE)
84      RETURN
85      300 ENTH = HYENTH (PE,TE)
86      RETURN
87      C
88      C      FIND DENSITY OF O2 OR H2 (LIG. OR GAS)
89      C
90      ENTRY FDNSTY (IG,TEMP,PRES,RHO)
91      C
92      IF (PRES .GT. PCRT(IG) .AND. TEMP .GT. CRTMP(IG)) GO TO 400
93      C      LIQUID STATE
94      CALL RHOLIG (TEMP,IG,RHO)
95      RETURN
96      C
97      ENTRY GSDNST (IG,TEMP,PRES,RHO)
98      C      DENSITY AND Z RETURNED
99      ENTRY GSZONS (IG,TEMP,PRES,RHO,Z)
100      C      GASEOUS STATE
101      400 CALL ZFIND (TEMP,PRES,IG,Z)
102      RHO = 144.*PRES / (Z*FINDR(IG)*TEMP)
103      RETURN
104      C
105      C      USED BY THE HEATEX ROUTINE
106      C
107      ENTRY TCRCLC (IGAS,FDPC,FPCIN,DPH,PHIN,TCR)
108      C
109      FDPOPC = FDPC / FPCIN
110      DPOPH = DPH / PHIN
111      C      CALC. TCR FOR O2 OR H2
112      490 TCR = CT1(IGAS)*(FPCIN**CT2(IGAS))/(PHIN**CT3(IGAS))
113      + (FDPOPC**CT4(IGAS))/(DPOPH**CT5(IGAS))
114      GO TO 500
115      C

```

```

***** MATHAX *****
116. ENTRY WOUACL (WOUA)
117. C
118. C      CALC. (W/U A)
119. C      ALPHW = CH4(IGAS)*FPCIN + CH5(IGAS)*PHIN
120. C
121. C      WOUA = CH1(IGAS)/((FPCIN**CH2(IGAS))*(PHIN**CH3(IGAS)))*EXP(ALPHW)
122. C      IS IT HYDROGEN
123. C      IF (IGAS .EQ. 1) GO TO 500
124. C      FOR H2 ONLY
125. C      WOUA = WOUA / ((FDPOPC**0.248)*(DPOPH**0.1656))
126. 500 CONTINUE
127. RETURN
128. C
129. C      ENTRY TCRLOW (FPCOUT,IGO)
130. C      LOWER TCR VALUE
131. C      FDPOPC = FDPOPC / 2.
132. C      IF (FDPOPC .GT. 0.001) GO TO 510
133. C      IGO = 2
134. C      FDPOPC = 0.001
135. C      GO TO 520
136. C
137. C      ENTRY TCRRAZ (FPCOUT,IGO)
138. C      RAISE TCR VALUE
139. C      FDPOPC = 2.0 * FDPOPC
140. C      IF (FDPOPC .LT. 0.20) GO TO 510
141. C      IGO = 2
142. C      FDPOPC = 0.20
143. C      GO TO 520
144. 510 IGO = 1
145. 520 FPCIN = FPCOUT / (1. - FDPOPC)
146. C      FDPC = FPCIN - FPCOUT
147. C      GO TO 490
148. C
149. END

```

***** FUNCTION MIPE

```

1 C *****
2 C ** ROUTINE NAME - MULTI-TABLE INTERPOLATION **
3 C ** AND POLYNOMIAL EVALUATION **
4 C ** ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2*
5 C ** PROGRAMMER - R. BOLLINGER 1943 102 26933 *
6 C ** DATE CODED - 3/3/70 *
7 C ** REVISED - SEPTEMBER 3, 1971 *
8 C ** PROGRAMMER - J. MCKAY 19-43 201 X45178 *
9 C *****
10 C
11 C FUNCTION MIPE(NL,XVAL)
12 C
13 C ***** EXPLANATION OF THE CALLING SEQUENCE *****
14 C
15 C NL - THE NUMBER OF LEVELS OF INTERPOLATION OR ONE
16 C LESS THAN THE NUMBER OF DIMENSIONS OR THE NUMBER
17 C OF VALUES IN THE ARRAY XVAR.
18 C
19 C XVAR - AN ARRAY OF INDEPENDENT VARIABLES ON WHICH
20 C TO INTERPOLATE OR EVALUATE POLYNOMIALS.
21 C
22 C LOGICAL DIAG
23 C
24 C REAL MIPE
25 C
26 C INCLUDE :CICOUNT
27 C INCLUDE :CKEYS
28 C INCLUDE :CTAB
29 C
30 C DIMENSION XVAL(6),YVAL(32),KNT(5),L(5),NTD(6),TAB1(6,5),
31 C ITAB1(6,5)
32 C
33 C EQUIVALENCE (TAB1,ITAB1,TAB,ITAB)
34 C
35 C ***** LOCATE THE POSITION IN EACH OF THE FIRST NL-1 TABLES OF
36 C ***** THE INDEPENDENT VARIABLE AND STORE THE INDEXES IN KNT.
37 C
38 C MPX = 1
39 C CONTINUE
40 C IF (DIAG(0,6HMIPE)) WRITE (107,6000) NL,(TAB1(I,I),I=1,5),
41 C (XVAL(I),I=1,NL)
42 C IF (NL.GT.1) GO TO 5
43 C CALL LOCAT (MPX)
44 C NLTB1 = 1
45 C CALL TEL(XVAL(1),YVAL(1))
46 C GO TO 220
47 C 5 NLMI = NL - 1
48 C DO 70 I1 = 1,NLMI
49 C NTAB1 = ITAB1(1,I1)
50 C IF (XVAL(I1) - TAB1(2,I1)) 50,50,10
51 C 10 IF (XVAL(I1) - TAB1(NTAB1+1,I1)) 20,60,60
52 C
53 C 20 DO 30 I2 = 1,NTAB1
54 C I3 = I2
55 C IF (XVAL(I1) - TAB1(I2+1,I1)) 40,40,30
56 C 30 CONTINUE
57 C

```



```

***** MIPE *****
58      40 KNT(11) = 13 - 1
59      GO TO 70
60      C
61      50 KNT(11) = 1
62      GO TO 70
63      C
64      60 KNT(11) = NTAB1 - 1
65      TO CONTINUE
66      C
67      C      ***** COMPUTE THE TABLE NUMBER OF THE XY TABLES FOR
68      C      ***** INTERPOLATION.
69      C
70      NT = 2**NLM1
71      NLTB1 = NT
72      DO 110 I1=1,NL
73      110 NTD(11) = NT / 2**(I1-1)
74      C      CALC. INTERPOLATION LEVEL COEF.
75      L(NLM1) = 1
76      IF (NLM1 .EQ. 0) GO TO 125
77      DO 120 I1=NLM1,2,-1
78      120 L(I1-1) = L(I1) * ITAB1(I1,I1)
79      C      COMPUTE THE ID NO. OF THE X-Y SUB-TABLES
80      C      USED FOR INTERPOLATION
81      125 DO 140 I1=1,NT
82      KTB = 1
83      DO 130 I2=1,NLM1
84      130 KTB = KTB + L(I2)* (KNT(I2)-1+MOD(I1-1,NTD(I2))/NTD(I2+1))
85      C      LOOK UP SUBTABLE LOCATION
86      CALL LOCAT (MPX*KTB)
87      C      COMPUTE THE YVAL FOR XY-CURVES OR
88      C      POLYNOMIALS
89      CALL TEL (XVAL(NL),YVAL(I1))
90      IF (KEY2 .EQ. KEY1) RETURN
91      140 CONTINUE
92      IF (DIAG(2,6HMIPE )) WRITE (107,6000) KNT,NT *(YVAL(I1),I1,NT)
93      C
94      C      ***** COMPUTE THE FINAL VALUE OF YVAL
95      C
96      DO 210 I1 = NLM1,1,-1
97      KYVAL = 1
98      IDXI = KNT(I1) + 1
99      FAC = (XVAL(I1) - TAB1(IDXI ,I1))/(TAB1(IDXI+1,I1)-TAB1(IDXI,I1))
100      C
101      DO 200 I2 = 1,NT,2
102      YVAL(KYVAL) = (YVAL(I2+1) - YVAL(I2))*FAC + YVAL(I2 )
103      KYVAL = KYVAL + 1
104      200 CONTINUE
105      C
106      NT = 2**(I1-1)
107      210 CONTINUE
108      C
109      C      ***** SET THE ANSWER AND RETURN TO THE CALLING PROGRAM
110      C
111      220 MIPE = YVAL(1)
112      IF (DIAG(1,6HMIPE )) WRITE (107,6000) NTD,YVAL(1)
113      RETURN
114      C
115      C      RYMIPE CAUSES INDEP AND DEP VAR TO SWITCH

```

```

***** HIPE *****
116 ENTRY RVHIPE (NL,XVAL)
117 C
118 MPX = -1
119 GO TO 1
120 C
121 6000 FORMAT (14X,17,5I15/(30X5E15,6))
122 END

```

***** FUNCTION NIENTH

```
1      FUNCTION NIENTH(PRES,TEMP)
2      C
3      C ROUTINE CALLS STEWARTS O2-N2 PROPERTIES PROGRAM
4      C
5      REAL NIENTH
6      C
7      CALL ONPROP(TEMP,PRES,DX,ENTH18,SX,UX,IX,18)
8      NIENTH = ENTH18
9      RETURN
10     END
```

***** SUBROUTINE ONPROP

```

1      SUBROUTINE ONPROP(TEMP,PRES,DENS,ENTH,ENTR,ENERG,ZEE,NGAS)
2      C
3      COMMON /METH/ M
4      C
5      C INPUT TO THIS SUBROUTINE MUST BE IN BRITISH UNITS
6      C
7      C
8      TB = TEMP
9      PB = PRES
10     DB = DENS
11     M = 1
12     C
13     IF(NGAS.EQ.1) KF = 1
14     IF(NGAS.EQ.18) KF = 2
15     C
16     KF = 1 CALL IN OXYGEN PARAMETERS
17     KF = 2 CALL IN NITROGEN PARAMETERS
18     C
19     IF(KF.EQ.1) CALL DATAO2
20     IF(KF.EQ.2) CALL DATAN2
21     C
22     IF((TB.GT.0.000).AND.(PB.GT.0.000)) GO TO 5
23     IF((TB.GT.0.000).AND.(DB.GT.0.000)) GO TO 10
24     C
25     5 CALL PROPB(TB,PB,DB,1,HB,SB,UB,ZB)
26     DENS = DB
27     ENTH = HB
28     ENTR = SB
29     ENERG = UB
30     ZEE = ZB
31     C
32     RETURN
33     C
34     10 CALL PROPB(TB,PB,DB,2,HB,SB,UB,ZB)
35     PRES = PB
36     ENTH = HB
37     ENTR = SB
38     ENERG = UB
39     ZEE = ZB
40     C
41     RETURN
42     C
43     END

```

SUBROUTINE ØTRTNS

```

1      C
2      C
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     6010 FORMAT(IH0,T39,**** SUMMARY OF COMPUTED HEAT EXCHANGER CHARACTERIS
26         TICS ****//T59,1FOR UNITS1,T73,A6,T93,A6)
27     6020 FORMAT(IH0,T29,**** SUMMARY OF COMPUTED HEAT EXCHANGER-GAS GENERAT
28         IOR CHARACTERISTICS ****)
29     6025 FORMAT(IH0,T36,**** SUMMARY OF COMPUTED WASTE HOT GAS UTILIZATION
30         IPARAMETERS ****)
31     6026 FORMAT(IH0,T36,**** SUMMARY OF COMPUTED FREON COOLANT UTILIZATION
32         IPARAMETERS ****)
33     6030 FORMAT(IH0,T35,**** SUMMARY OF COMPUTED HEAT EXCHANGER CHARACTERIS
34         TICS -- (CONTINUED ****)
35     6040 FORMAT(IH0,T40,**** SUMMARY OF COMPUTED PUMP CHARACTERISTICS FOR T
36         HE SYSTEM ****)
37     6050 FORMAT(IH0,T36,**** SUMMARY OF COMPUTED TURBINE CHARACTERISTICS FO
38         IR THE SYSTEM ****)
39     6060 FORMAT(IH0,T36,**** SUMMARY OF COMPUTED TURBINE GAS GENERATOR CHAR
40         IACTERISTICS ****)
41     6065 FORMAT(///T46,**** ECLSS ENERGY REQUIREMENTS SUMMARY *** i)
42     6070 FORMAT(IH0,T46,**** INITIAL TANK SIZING CALCULATIONS ****)
43     6080 FORMAT(IH0,T47,**** FINAL TANK SIZING CALCULATIONS ****)
44     6090 FORMAT(IH0,T47,**** ACCUMULATOR SIZING CALCULATIONS ****)
45     6100 FORMAT(IH0,T39,**** TANK PROPELLANT ACQUISITION DEVICE COMPUTATION
46         I ****)
47     6110 FORMAT(IH0,T51,**** COMPONENT WEIGHT SUMMARY ****// T27,1... OXIDY
48         IZER ...T87,1... FUEL ... / T31,1-----T91,1-----/ T38,1COMPON
49         ZENT1T52,1INSULATION1T98,1COMPONENT1T112,1INSULATION1 / T11,1COMPON
50         3ENT1T27,1CODE1T38,1WT. (LBS)1T52,1WT. (LBS)1T71,1COMPONENT1T87,
51         41CODE1T98,1WT. (LBS)1T112,1WT. (LBS)1/)
52     6112 FORMAT(IH0,T47,**** COMPONENT WEIGHT SUMMARY TOTALS ****)
53     6120 FORMAT(IH0,T52,**** COMPUTED FLOWRATE DATA ****)
54     6130 FORMAT(IH0,T47,**** INITIAL APU PROGRAM CALCULATIONS ****//
55         1 T40,1PERCENT1T61,1TOT. FLOWRATE1T85,1EXHAUST TEMP.1 /
56         2 T25,1CYCLE1T42,1H.P.1T63,1(LB./MIN.)1T87,1(DEG. - R)1/)
57     6132 FORMAT(T46,**** APU MIXTURE RATIO IS NOW SET AT1,F5.2,1 ****//)

```

***** OTRTNS *****

```

58 6140 FORMAT(24X14,E20.6,2E24.6)
59 6150 FORMAT(1H0,T51,1*** APU CALCULATIONS - CONT. ***1)
60 6160 FORMAT(T44,1*** APU MIXTURE RATIO IS NOW SET AT F5.2,1 ***1///
61 1 T47,1*** APU SUPERCRITICAL CALCULATIONS ***1)
62 6161 FORMAT(4X12,8E15.6)
63 6162 FORMAT(1H0 T14,1CONDITIONING GAS RECD,T44,1WGT,1CONDITIONING GAS
64 1 T74,1CONDITIONING GAS RECD,T104,1WGT,1CONDITIONING GAS
65 2 /T2,1CYCLE,T15,1ACCUMULATOR TO APU, T45,1ACCUMULATOR TO APU,
66 3 T75,1TANK TO ACCUMULATOR, T105,1TANK TO ACCUMULATOR,
67 4 / T6,4(8X1OXYGEN'8X1HYDROGEN'1))
68 6164 FORMAT(1H0 T14,1CONDITIONING GAS RECD,T44,1WGT,1CONDITIONING GAS
69 1 T74,1FLOW TO ACCUMULATOR, T101,1EXHAUST PROD,1
70 2 /T2,1CYCLE,T21,1FOR TANK, T51,1FOR TANK,T78,1(EACH CYCLE),1
71 3 T103,1SURPLUS,T6,3(8X1OXYGEN'8X1HYDROGEN'1))
72 6166 FORMAT(1H0 T16,1REFERENCE FLOW TO, T44,1TEMPERATURE OF FLUID,1
73 1 T75,1PERCENT OF USABLE, T104,1HEAT XFR. INTO TANK,1
74 2 /T2,1CYCLE,T15,1SUPPLEMENTAL G. G., T46,1IN STORAGE TANK,1
75 3 T76,1FLUID WITHDRAWN, T108,1(EACH CYCLE),1
76 4 / T6,4(8X1OXYGEN'8X1HYDROGEN'1))
77 6168 FORMAT(1H0 T16,1ENERGY DERIVATIVE, T44,1COR. COND. GAS RECD,1
78 1 T74,1COR. WGT. COND. GAS,1
79 2 /T2,1CYCLE,T45,1TANK TO ACCUMULATOR, T74,1TANK TO ACCUMULATOR,1
80 3 / T6,3(8X1OXYGEN'8X1HYDROGEN'1))
81 6170 FORMAT(1H0 T44,1COR. COND. GAS RECD,1 T74,1COR. WGT. COND. GAS,1
82 1/T32,1CYCLE,T50,1FOR TANK, T80,1FOR TANK,1
83 2 / T36,2(8X1OXYGEN'8X1HYDROGEN'1))
84 6171 FORMAT(34X12,6E15.6)
85 6172 FORMAT(T44,1*** APU MIXTURE RATIO IS NOW SET AT F5.2,1 ***1///
86 1 T48,1*** APU SUBCRITICAL CALCULATIONS ***1)
87 6174 FORMAT(1H0 T14,1CONDITIONING GAS RECD,T44,1WGT,1CONDITIONING GAS
88 1 T74,1CONDITIONING GAS RECD,T104,1WGT,1CONDITIONING GAS
89 2 /T2,1CYCLE,T15,1ACCUMULATOR TO APU, T45,1ACCUMULATOR TO APU,
90 3 T75,1PUMP TO ACCUMULATOR, T105,1PUMP TO ACCUMULATOR,
91 4 / T6,4(8X1OXYGEN'8X1HYDROGEN'1))
92 C
93 NIENH = 0.0
94 C DO OUTPUT FOR ALL HEAT EXCHANGERS
95 C
96 IDH2 = 2*IDMHX
97 IDH3 = 3*IDMHX
98 IDH6 = 6*IDMHX
99 DO 140 JX=1,NUMHEX
100 KVL = 0
101 IF (NSUK(JX,1).EQ.0) KVL = KVL + 1
102 IF ( NSUK(JX,2).EQ.0) KVL = KVL + 2
103 MVL = 3 - KVL
104 IF (MVL.EQ.0) GO TO 140
105 IF (PAGE( 0)) WRITE (107,6010) UCODE(JX,1),UCODE(JX,2)
106 JP = PAGE (17)
107 CALL SPACE
108 CALL OUTPA (5,LHX3,MVL,JFLUID,JFLUID(1,2))
109 CALL SPACE
110 C
111 DO 100 I2=1,11
112 JSM = (I2-1)*IDH2 + JX
113 JLM = JSM + IDMHX
114 C
115 CALL OUTPF (4,LHX2(1,I2),MVL,UOTHX(JSM),UOTHX(JLM))

```

***** OTRTNS *****

```

116      100 CONTINUE
117      J1 = NSSK(JX,1) - 1
118      J2 = NSSK(JX,2) - 1
119      MXNSU = NSUK(JX,1)
120      IF (MXNSU .LT. NSUK(JX,2)) MXNSU = NSUK(JX,2)
121      C
122      DO 120 I2=1,MXNSU
123      IF (PAGE(1)) WRITE (IOT,6030)
124      J1 = J1 + 1
125      K1 = 1
126      IF (J1 .EQ. 1 .AND. IISU(JX,1) .GT. 7) K1 = 0
127      J2 = J2 + 1
128      K2 = 1
129      IF (J2 .EQ. 1 .AND. IISU(JX,2) .GT. 7) K2 = 0
130      NVL = 0
131      IF (J1 .LE. NSSK(JX,1)+NSUK(JX,1)-1) NVL = NVL+1
132      IF (J2 .LE. NSSK(JX,2)+NSUK(JX,2)-1) NVL = NVL+2
133      CALL SPACE
134      CALL OUTPA (4,LHX1(1,10),NVL,UNAM(1,J1+K1),UNAM(1,J2+K2))
135      CALL SPACE
136      DO 110 I3=1,8
137      JSM = (I3-1)*IDM6 + (J1-1)*IDMHX + JX
138      JLM = (I3-1)*IDM6 + (J2-1)*IDMHX + IDM3 + JX
139      C
140      CALL OUTPF (4,LHX1(1,I3),NVL,UOTHY(JSM),UOTHY(JLM))
141      110 CONTINUE
142      120 CONTINUE
143      IF (PAGE(2)) WRITE (IOT,6030)
144      CALL SPACE
145      C
146      CALL OUTPF (4,LHX1(1,9),MVL,WHXTOT(JX,1),WHXTOT(JX,2))
147      JP = PAGE (10)
148      C
149      JSTYP = HSTYPE(JX,2)
150      IF (JSTYP.EQ.2) GO TO 125
151      C
152      WRITE (IOT,6020)
153      CALL SPACE
154      C
155      CALL OUTPA (5,LHS1,MVL,JFLUID,JFLUID(1,2))
156      CALL SPACE
157      CALL OUTPF (5,LHS1(1,2),MVL,WDOTH(JX,1),WDOTH(JX,2))
158      CALL OUTPF (5,LHS1(1,3),MVL,WGGFX(JX,1),WGGFX(JX,2))
159      CALL OUTPF (5,LHS1(1,4),MVL,HSWGHT(JX,1),HSWGHT(JX,2))
160      CALL SPACE
161      CALL OUTPF (5,LHS1(1,6),MVL,HXASSY(JX,1),HXASSY(JX,2))
162      C
163      IF (JSTYP .EQ. 1) GO TO 140
164      C
165      125 CONTINUE
166      WRITE (IOT,6025)
167      CALL SPACE
168      C
169      CALL OUTPA (5,LHS2(1,6),MVL,JFLUID,JFLUID(1,2))
170      CALL SPACE
171      CALL OUTPF (5,LHS2(1,1),MVL,MSGCPE(JX,1),MSGCPE(JX,2))
172      CALL OUTPF (5,LHS2(1,2),MVL,MSGREQ(JX,1),MSGREQ(JX,2))
173      CALL OUTPF (5,LHS2(1,3),MVL,MSGTOT(JX,1),MSGTOT(JX,2))

```

```

***** OTRNS *****
174 C
175 140 CONTINUE
176 IF (PAGE ( 5)) WRITE (IOT,6030)
177 CALL SPACE
178 CALL OUTPF (5,LHS1(1,5),3,WGGTOT,WGGTOT(2))
179 C
180 CALL SPACE
181 CALL OUTPF (5,LHS2(1,4),3,GTREG(1),GTREG(2))
182 CALL OUTPF (5,LHS2(1,5),3,HFTOT(1),HFTOT(2))
183 RETURN
184 C
185 ENTRY OTRHXF
186 C
187 *** OUTPUT FUEL CELL HEAT EXCHANGER PARAMETERS
188 C
189 IDM2 = 2*IDMHX
190 DO 141 JX = 1,NUMHEX
191 IF (PAGE(0)) WRITE(IOT,6010) UCODE(JX,1),UCODE(JX,2)
192 JP = PAGE (17)
193 MVL = 3
194 CALL SPACE
195 CALL OUTPA (5,LHX3,MVL,JFLUID,JFLUID(1,2))
196 CALL SPACE
197 C
198 DO 142 I2 = 1,8
199 JSM = (I2-1) * IDM2 + JX
200 JLM = JSM + IDMHX
201 CALL OUTPF (4,LHX2(1,I2),MVL,UOTHX(JSM),UOTHX(JLM))
202 142 CONTINUE
203 CALL SPACE
204 CALL OUTPF (4,LHX1(1,9),MVL,WHXTOT(JX,1),WHXTOT(JX,2))
205 C
206 WRITE (IOT,6026)
207 CALL SPACE
208 CALL OUTPA (5,LHS2(1,6),MVL,JFLUID,JFLUID(1,2))
209 CALL SPACE
210 CALL OUTPF (5,LHS2(1,1),MVL,HSGCPE(JX,1),HSGCPE(JX,2))
211 CALL OUTPF (5,LHS2(1,2),MVL,HSGREG(JX,1),HSGREG(JX,2))
212 CALL OUTPF (5,LHS2(1,3),MVL,HSGTOT(JX,1),HSGTOT(JX,2))
213 141 CONTINUE
214 C
215 CALL SPACE
216 CALL OUTPF (5,LHS2(1,4),3,GTREG(1),GTREG(2))
217 CALL OUTPF (5,LHS2(1,7),3,WOTFMX ,WOTFMX )
218 C
219 RETURN
220 C
221 C
222 ENTRY OTRHXE
223 C
224 *** OUTPUT THE ECLSS HEAT EXCHANGER PARAMETERS
225 C
226 IDM2 = 2*IDMHX
227 DO 143 JX = 1,NUMHEX
228 IF (PAGE(0)) WRITE(IOT,6010) UCODE(JX,1),UCODE(JX,2)
229 JP = PAGE (17)
230 MVL = 3
231 CALL SPACE

```



```

***** OTRTNS *****
232 CALL OUTPA(5,LHX3,MVL,JFLUID,JFLUID(1,3))
233 CALL SPACE
234 DO 144 I2 = 1,2
235 JSM = (I2-1) * IDM2 + JX
236 JLM = JSM + IDMHX
237 CALL OUTPF (4,LHX2(1,12),MVL,UOTHX(JSM),UOTHX(JLM))
238 144 CONTINUE
239 C
240 CALL OUTPF(4,LHX2(1,4),MVL,WDOTCF(JX,1),WDOTCF(JX,2))
241 CALL OUTPF (4,LHX2(1,12),MVL,UOA(JX,1),UOA(JX,2))
242 CALL OUTPF (4,LHX2(1,13),MVL,DH(JX,1),DH(JX,2))
243 CALL OUTPF (4,LHX2(1,14),MVL,HLNGTH(JX,1),HLNGTH(JX,2))
244 CALL SPACE
245 CALL OUTPF(4,LHX1(1,3),MVL,HXCDLP(JX,1),HXCDLP(JX,2))
246 CALL SPACE
247 CALL OUTPF (4,LHX1(1,9),MVL,WHXTOT(JX,1),WHXTOT(JX,2))
248 CALL SPACE
249 CALL OUTPF (5,LHS2(1,2),MVL,HSQREG(JX,1),HSQREG(JX,2))
250 143 CONTINUE
251 C
252 RETURN
253 C
254 ENTRY OPTPOW
255 C
256 *** OUTPUT THE POWER REQUIREMENTS SUMMARY
257 C
258 WRITE (IOT,6065)
259 CALL SPACE
260 CALL OUTPFI (5,LHS2(1,8),HWTOMX)
261 CALL OUTPFI (5,LHS2(1,9),HWTNMX)
262 CALL OUTPFI (5,LHS2(1,10),TWTOMX)
263 CALL OUTPFI (5,LHS2(1,11),TWTNMX)
264 CALL SPACE
265 CALL OUTPFI (5,LHS2(1,12),TOTWMX)
266 CALL SPACE
267 CALL OUTPFI (5,LHS2(1,13),TOTWAT)
268 CALL SPACE
269 CALL OUTPFI (5,LHS2(1,14),TOTPOW)
270 C
271 RETURN
272 C
273 C
274 ENTRY OTPPHP
275 C
276 OUTPUT PUMP PARAMETERS
277 IF (PAGE(0)) WRITE (IOT,6040)
278 JP = PAGE (21)
279 CALL SPACE
280 CALL OUTPA (4,LPP5,3,JFLUID,JFLUID(1,2))
281 CALL SPACE
282 DO 200 I=1,6
283 I2 = 2*(I-1)+1
284 CALL OUTPF (3,LPP1(1,I),3,UOTPI(I2),UOTPI(I2+1))
285 200 CONTINUE
286 CALL SPACE
287 CALL OUTPI (4,LPP2,3,PSTAGE,PSTAGE(2))
288 DO 210 I=1,6
289 I2 = 2*(I-1)+1
290 CALL OUTPF (3,LPP3(1,I),3,UOTP2(I2),UOTP2(I2+1))

```

```

***** OTRTNS *****
290      210 CONTINUE
291      CALL SPACE
292      CALL OUTPI (4,LPP4,3,JJOPT,JJOPT(2))
293      RETURN
294      C
295      ENTRY OTPTRB
296      C          DO OUTPUT FOR TURBINES
297      JP = PAGE (19)
298      WRITE (IOT,6050)
299      CALL SPACE
300      CALL OUTPA (4,LTBN1,3,JFLUID,JFLUID(1,2))
301      CALL SPACE
302      DO 300 I=1,6
303      I2 = 2*(I-1)+1
304      CALL OUTPF (5,LTBN2(1,I),3,UOTB1(I2),UOTB1(I2+1))
305      300 CONTINUE
306      WRITE (IOT,6060)
307      CALL SPACE
308      CALL OUTPA (5,LHS1,3,JFLUID,JFLUID(1,2))
309      CALL SPACE
310      DO 310 I=1,3
311      I2 = 2*(I-1)+1
312      CALL OUTPF (5,LHS1(1,I+1),3,UOTB2(I2),UOTB2(I2+1))
313      310 CONTINUE
314      RETURN
315      C
316      ENTRY OTPTSZ (IFLG)
317      C          OUTPUT TANK SIZING PARAMETERS
318      JP = PAGE (0)
319      JP = PAGE (17)
320      IF (IFLG.EQ. 2) GO TO 400
321      WRITE (IOT,6070)
322      GO TO 410
323      400 WRITE (IOT,6080)
324      410 CALL SPACE
325      CALL OUTPA (1,6H      ,3,JFLUID,JFLUID(1,2))
326      CALL SPACE
327      CALL OUTPI (3,LTZI(1,1),3,NOP ,NOP (2,1))
328      CALL OUTPI (3,LTZI(1,2),3,SMTYPE,SMTYPE(2,1))
329      IF (IFLG.EQ. 2)
330      I CALL OUTPI (3,LTZI(1,3),3,SITYPE,SITYPE(2,1))
331      CALL OUTPF (3,LTZI(1,4),3,WPTOT ,WPTOT (2,1))
332      CALL OUTPF (3,LTZI(1,5),3,SVLFLO,SVLFLO(2,1))
333      CALL OUTPF (3,LTZI(1,14),3,TCYHT,TCYHT (2,1))
334      CALL OUTPF (3,LTZI(1,6),3,SMDIAM,SMDIAM(2,1))
335      CALL OUTPF (3,LTZI(1,7),3,TSA ,TSA (2,1))
336      CALL OUTPF (3,LTZI(1,8),3,SVOL ,SVOL (2,1))
337      CALL OUTPF (3,LTZI(1,9),3,TWT ,TWT (2,1))
338      IF (IFLG.EQ. 2) GO TO 420
339      CALL OUTPF (3,LTZI(1,12),3,SHRATE,SHRATE(2,1))
340      GO TO 430
341      420 CALL OUTPF (3,LTZI(1,10),3,SITHIK,SITHIK(2,1))
342      CALL OUTPF (3,LTZI(1,11),3,TINT ,TINT (2,1))
343      C
344      430 RETURN
345      C
346      ENTRY OTPACC
347      C          OUTPUT ACCUMULATOR DATA

```

```

***** OTRTNS *****
348      IF (PAGE( 0)) WRITE (107,6090)
349      JP = PAGE (16)
350      CALL SPACE
351      CALL OUTPA (1,6H      ,3,JFLUID,JFLUID(1,2))
352      CALL SPACE
353      DO 500 I=1,3
354      I2 = 2*(I-1)+1
355      500 CALL OUTPI (3,LTZI(1,I),3,EGAC(I2),EGAC(I2+1))
356      CALL OUTPF (3,LTZI(1,14),3,ACYHT,ACYHT(2))
357      DO 510 I=6,11
358      I2 = 2*(I-3)+1
359      510 CALL OUTPF (3,LTZI(1,I),3,EGAC(I2),EGAC(I2+1))
360      CALL OUTPF (3,LTZI(1,13),3,WGRACC,WGRACC(2))
361      RETURN
362
363      C      ENTRY OTPACQ
364      C      OUTPUT ACQUISITION DEVICE DATA
365      IF ( PAGE( 0)) WRITE (107,6100)
366      JP = PAGE ( 9)
367      CALL SPACE
368      CALL OUTPA (1,6H      ,3,JFLUID,JFLUID(1,2))
369      CALL SPACE
370      I1 = SATYPE
371      I2 = SATYPE(2)
372      CALL OUTPA (3,LTZ3(1,1),3,LTZ2(1,I1),LTZ2(1,I2))
373      CALL OUTPF (3,LTZ3(1,2),3,WTAC0,WTAC0(2,1))
374      CALL OUTPF (3,LTZ3(1,3),3,WLR,WLR(2))
375      CALL OUTPF (3,LTZ3(1,4),3,WLRT,WLRT(2,1))
376      RETURN
377
378      C      ENTRY OTPWSH
379      C      OUTPUT WEIGHT SUMMARY DATA
380      IF (PAGE( 0)) WRITE (107,6110)
381      JP = PAGE ( 8)
382      I1 = IOSTT
383      I2 = IHSTT
384      KH2 = 1
385      KO2 = 1
386      MVL = 3
387      NCPH = KHEND - IHSTT
388      NCPO = KOEND - IOSTT
389      MXCP = NCPO
390      IF (MXCP .LT. NCPH) MXCP = NCPH
391      H2IWT = 0.
392      H2SWT = 0.
393      O2IWT = 0.
394      O2SWT = 0.
395
396      C      DO 770 I=1,MXCP
397      GO TO (700,730),KO2
398
399      C      SET UP OXIDIZER SIDE
700      I1 = I1 + 1
400      IF (I1 .LE. KOEND) GO TO 710
401      KO2 = 2
402      MVL = MVL - 1
403      GO TO 730
404      710 CALL GETCON (I1)
405      IF (CFUNCT.EQ. 1) GO TO 700

```

```

***** OTRNS *****
406      KONAM = FNAME(CFUNCT)
407      O2IWT = O2IWT + WI(I)
408      O2SWT = O2SWT + WEIGHT(I)
409
410      C
411      730 GO TO (740,760),KH2
412      C      SET UP FUEL SIDE
413      740 I2 = I2 + 1
414      IF (I2 .LE. KHEND) GO TO 750
415      KH2 = 2
416      MVL = MVL - 2
417      GO TO 760
418      750 CALL GETCON (I2)
419      IF (CFUNCT .EQ. 1) GO TO 740
420      KHNAM = FNAME(CFUNCT)
421      H2IWT = H2IWT + WI(I2)
422      H2SWT = H2SWT + WEIGHT(I2)
423
424      C
425      760 IF (MVL .EQ. 0) GO TO 780
426      IF (PAGE(1))WRITE (IOT,6110)
427      PRINT A LINE
428      CALL OUTPW (MVL,KONAM,CODE(I1),WEIGHT(I1),WI(I1),
429      KHNAM,CODE(I2),WEIGHT(I2),WI(I2))
430      770 CONTINUE
431
432      C
433      780 CONTINUE
434      JP = PAGE ( 9)
435      CALL SPACE
436      WRITE (IOT,6112)
437      TTLSWT = ENGHT + H2SWT + H2IWT + O2SWT + O2IWT
438      CALL SPACE
439
440      C      PRINT SYSTEM WT. TOTALS
441      CALL OUTPFI (4,LCNFI,ENGHT)
442      DO 790 I=1,5
443      CALL OUTPFI (4,LCNFI(I,I+1),WTOFSY(I))
444      790 CONTINUE
445      RETURN
446
447      C
448      ENTRY OTPELT
449
450      C      OUTPUT FLOWRATE DATA (FOR TURB.GG)
451      IF (PAGE( 0)) WRITE (IOT,6120)
452      JP = PAGE (15)
453      CALL SPACE
454      JKM = 0
455      I1 = 1
456      I2 = 3
457      800 CALL OUTPA (1,6H      ,3,KFLUID,KFLUID(1,2))
458      810 CALL SPACE
459
460      C
461      DO 820 I=1,I2
462      820 CALL OUTPF (3,LFRT(I,I),3,EGRT(I,1),EGRT(I,2))
463      JKM = JKM + 1
464      GO TO (830,840,840),JKM
465
466      C
467      ENTRY OTPELX
468
469      C      (FOR HEX GG FLOWRATES)
470      JP = PAGE (11)
471      CALL SPACE
472      WRITE (IOT,6120)

```

```

***** OTRTNS *****
464      JKM = 2
465      C
466      830 CALL SPACE
467          I1 = 4
468          I2 = 6
469          GO TO (810,800),JKM
470      840 CALL SPACE
471          CALL OUTPF (3,LFRT(1,7),9,WDOYT,WDOYT(2))
472          RETURN
473      C
474          ENTRY OPAPUF (NCY)
475      C          OUTPUT APU FLOWRATE DATA
476          IF (PAGE( 0)) WRITE (IOT,6132) FMR
477          WRITE (IOT,6130)
478          JP = PAGE (6+NCY)
479          DO 900 I=1,NCY
480      900 WRITE (IOT,6140) I,PCTHP(I),WD(I),TE(I)
481          IF (PAGE ( 9)) WRITE (IOT,6150)
482          CALL SPACE
483          CALL SPACE
484          DO 910 I=1,5
485      910 CALL OUTPF1 (4,LAPUI(1,I),EGAPI(1))
486          CALL OUTPF1 (4,LAPUI(1,6),WDOI )
487          CALL OUTPF1 (4,LAPUI(1,7),WDOI(2))
488          RETURN
489      C
490          ENTRY OAPUSP (NCY)
491      C          OUTPUT APU-SUPERCRITICAL DATA
492          IF (PAGE( 0)) WRITE (IOT,6160) FMR
493          JP = PAGE (9+NCY)
494      C
495          WRITE (IOT,6162)
496          DO 1000 I=1,NCY
497      1000 WRITE (IOT,6161) I,Q1ODOT(I),Q1HDOT(I),WDD(I),WDA(I),
498          |          Q2ODOT(I),Q2HDOT(I),WDE(I),WDR(I)
499      C
500          IF (PAGE (7+NCY)) WRITE (IOT,6160) FMR
501          WRITE (IOT,6164)
502          DO 1010 I=1,NCY
503      1010 WRITE (IOT,6161) I,Q3ODOT(I),Q3HDOT(I),WDF(I),WDC(I),
504          |          WTD(I),WTH(I),DWDB(I)
505      C
506          IF (PAGE (7+NCY)) WRITE (IOT,6160) FMR
507          WRITE (IOT,6166)
508          DO 1020 I=1,NCY
509      1020 WRITE (IOT,6161) I,WGOC(I),WGHC(I),TTO(I),TTH(I),PCO2WD(I),
510          |          PCH2WD(I),DRODWD(I),DOODWH(I)
511      C
512          IF (PAGE (7+NCY)) WRITE (IOT,6160) FMR
513          WRITE (IOT,6168)
514          DO 1030 I=1,NCY
515      1030 WRITE (IOT,6161) I,PHI02(I),PHI12(I),Q2ODTC(I),Q2HDTC(I),
516          |          WDEC(I),WDBC(I)
517      C
518          IF (PAGE (7+NCY)) WRITE (IOT,6160) FMR
519          WRITE (IOT,6170)
520          DO 1040 I=1,NCY
521      1040 WRITE (IOT,6171) I,Q3ODTC(I),Q3HDTC(I),WDFC(I),WDCC(I)

```

***** OTRTNS *****

```

522 C
523 IF (PAGE( 0)) WRITE (IOT,6160) FMR
524 JP = PAGE (18)
525 CALL SPACE
526 CALL OUTPA (1,6H ,3,JFLUID,JFLUID(1,2))
527 CALL SPACE
528 DO 1050 I=1,11
529 1050 CALL OUTPF (4,LAPU2(1,I),3,EGAP2(1,I),EGAP2(1,2))
530 RETURN
531 C
532 ENTRY OAPUSB (NCY)
533 C OUTPUT APU = SUBCRITICAL DATA
534 IF (PAGE( 0)) WRITE (IOT,6172) FMR
535 JP = PAGE (9+NCY)
536 C
537 WRITE (IOT,6174)
538 DO 1100 I=1,NCY
539 1100 WRITE (IOT,6161) I,Q60DOT(I),Q4H0OT(I),WDJ(I),WDG(I),
540 Q70DOT(I),Q5HDOT(I),WGG0(I),WGGH(I)
541 C
542 IF (PAGE( 0)) WRITE (IOT,6172) FMR
543 JP = PAGE (18)
544 CALL SPACE
545 CALL OUTPA (1,6H ,3,JFLUID,JFLUID(1,2))
546 CALL SPACE
547 C
548 DO 1110 I=1,11
549 NVL = 3
550 IF (I.EQ. 7) NVL = 2
551 1110 CALL OUTPF (4,LAPU3(1,I),NVL,EGAP3(I,1),EGAP3(I,2))
552 RETURN
553 C
554 END

```

***** SUBROUTINE OUTPUT

```

1      C
2      SUBROUTINE OUTPF (NTL,ITITL,NVL,VAR1,VAR2)
3      C
4      DIMENSION ITITL(6),IVA1(2),IVA2(2)
5      C
6      6001 FORMAT (29X,6A6)
7      6002 FORMAT ('+',64X,E15.6)
8      6003 FORMAT ('+',84X,E15.6)
9      6004 FORMAT ('+',64X,I11)
10     6005 FORMAT ('+',84X,I11)
11     6006 FORMAT ('+',68X,2A6)
12     6007 FORMAT ('+',88X,2A6)
13     6008 FORMAT (' ')
14     6009 FORMAT (T13,A6,T27,A6,2F14.3,T66,' ',I)
15     6010 FORMAT (T66,' ',T73,A6,T87,A6,2F14.3)
16     6011 FORMAT (T13,A6,T27,A6,2F14.3,T66,' ',T73,A6,T87,A6,2F14.3)
17     C
18     WRITE (IOU,6001) (ITITL(I),I=1,NTL)
19     GO TO (110,120,110),NVL
20     110 WRITE (IOU,6002) VAR1
21     IF (NVL.EQ. 1) RETURN
22     120 WRITE (IOU,6003) VAR2
23     RETURN
24     C
25     ENTRY OUTPFI (NTL,ITITL,VAR1)
26     C
27     WRITE (IOU,6001) (ITITL(I),I=1,NTL)
28     WRITE (IOU,6002) VAR1
29     RETURN
30     C
31     ENTRY OUTPI (NTL,ITITL,NVL,IVR1,IVR2)
32     C
33     WRITE (IOU,6001) (ITITL(I),I=1,NTL)
34     GO TO (210,220,210),NVL
35     210 WRITE (IOU,6004) IVR1
36     IF (NVL.EQ. 1) RETURN
37     220 WRITE (IOU,6005) IVR2
38     RETURN
39     C
40     ENTRY OUTPA (NTL,ITITL,NVL,IVA1,IVA2)
41     C
42     WRITE (IOU,6001) (ITITL(I),I=1,NTL)
43     GO TO (310,320,310),NVL
44     310 WRITE (IOU,6006) IVA1
45     IF (NVL.EQ. 1) RETURN
46     320 WRITE (IOU,6007) IVA2
47     RETURN
48     C
49     ENTRY OUTPW (NVL,JVA1,JVA2,VAR1,VAR3,JVA3,JVA4,VAR2,VAR4)
50     C
51     GO TO (410,420,430),NVL
52     410 WRITE (IOU,6009) JVA1,JVA2,VAR1,VAR3
53     GO TO 450
54     420 WRITE (IOU,6010) JVA3,JVA4,VAR2,VAR4
55     GO TO 450
56     430 WRITE (IOU,6011) JVA1,JVA2,VAR1,VAR3,JVA3,JVA4,VAR2,VAR4
57     450 RETURN

```

```

*****  OUTPUT  *****
58      C
59      ENTRY SPACE
60      C
61      WRITE (IOU,6008)
62      RETURN
63      C
64      ENTRY OTUNIT (IONO)
65      C
66      IOU = IONO
67      RETURN
68      END

```


***** FUNCTION OXENTH

```
1 FUNCTION OXENTH(PRES,TEMP)
2
3 C
4 C ROUTINE CALLS STEWARTS COMBINED O2-N2 PROPERTIES PROGRAM
5
6 CALL ONPROP(TEMP,PRES,DX,ENTHI,SX,UX,ZX,1)
7 OXENTH = ENTHI
8 RETURN
9 END
```

FUNCTION PAGE

1	C	*****	PAGE 0100001
2	C	* ROUTINE NAME = PAGE HEADING SUBFUNCTION.	PAGE 0200002
3	C	* ROUTINE LANG = FORTRAN IV UNIVAC 1107/1108 *	PAGE 0300003
4	C	* PROGRAMMER = R. BOLLINGER 5432 104 22898 *	PAGE 0400004
5	C	* DATE CODED = 03/10/66	PAGE 0500005
6	C	*****	PAGE 0600006
7	C		PAGE 0700007
8		FUNCTION PAGE(NLINES)	PAGE 0800008
9	C		PAGE 0900009
10	C	***** EXPLANATION OF CALLING SEQUENCE *****	PAGE 1000010
11	C		PAGE 1100011
12	C	INPUT TO PAGE	PAGE 1200012
13	C		PAGE 1300013
14	C	NLINES = NUMBER OF LINES	PAGE 1400014
15	C	IF NLINES = -1, THE ROUTINE WILL INITIALIZE	PAGE 1500015
16	C	THE ROUTINE FOR A NEW CASE,	PAGE 1600016
17	C	PAGE EJECT, AND OUTPUT THE	PAGE 1700017
18	C	PAGE HEADING.	PAGE 1800018
19	C	0, THE ROUTINE WILL PAGE EJECT,	PAGE 1900019
20	C	AND OUTPUT THE PAGE HEADING.	PAGE 2000020
21	C	*, THE ROUTINE WILL ADD NLINES	PAGE 2100021
22	C	TO LINECT AND TEST TO SEE IF	PAGE 2200022
23	C	LINECT IS GREATER THAN MAXLIN.	PAGE 2300023
24	C	IF IT IS, THE ROUTINE WILL	PAGE 2400024
25	C	PAGE EJECT AND OUTPUT THE PAGE	PAGE 2500025
26	C	HEADING.	PAGE 2600026
27	C		PAGE 2700027
28	C	OUTPUT FROM PAGE	PAGE 2800028
29	C		PAGE 2900029
30	C	PAGE = PAGE EJECT OR NO PAGE EJECT	PAGE 3000030
31	C	IF PAGE = TRUE, THE ROUTINE HAS PAGE EJECTED.	PAGE 3100031
32	C	FALSE, THE ROUTINE HAS NOT PAGE	PAGE 3200032
33	C	EJECTED.	PAGE 3300033
34	C		
35		LOGICAL PAGE	
36	C		PAGE 3400036
37		INCLUDE CPAGE	
38	C		PAGE 4100038
39		PAGE = .FALSE.	PAGE 4200039
40		IF(NLINES)10,30,20	PAGE 4300040
41	C		PAGE 4400041
42		10 NCASE = NCASE + 1	PAGE 4500042
43		CALL DATE(9,DOR)	PAGE 4600043
44		GO TO 40	PAGE 4900044
45	C		PAGE 5000045
46		20 LINECT = LINECT + NLINES	PAGE 5100046
47		IF(LINECT,LE,MAXLIN) RETURN	PAGE 5200047
48		30 LINECT = NLINES	PAGE 5400048
49		40 NPAGE = NPAGE + 1	PAGE 5410049
50		CALL TOD(8,TIME)	PAGE 5500050
51		PAGE = .TRUE.	PAGE 5700051
52		WRITE(OPTLUN,6000)NAME,NPAGE, DEPT,PTITLE,DOR,EXT,TIME,BLD,	PAGE 5800052
53		JNUM,NCASE,CTITLE	
54		RETURN	PAGE 5900054
55		6000 FORMAT (11,T28,NAME,1,2A6,21(1*),1,PAGE,1,15,/,	
56	1	T28,DEPT,1,A4,9X,1,6A6,1,*,DATE,1A6,A3,/,	
57	2	T28,EXT,1,A5,8X,1,39X,1,TIME,1,A6,A2,/,	

B-239

LMSC-A991396

***** PAGE *****

58 3 T28, 'BLD. ' A9, 10X! *' 17XA6, 16X ** CASE 'IS /
59 4 T28, 39(' * ') / T30, 12A6)
60 C
61 END

PAGE 6000060
PAGE 6500061

***** SUBROUTINE PARPMP

```

1      C
2      C
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C
40     C
41     C
42     C
43     C
44     C
45     C
46     C
47     C
48     C
49     C
50     C
51     C
52     C
53     C
54     C
55     C
56     C
57     C

```

***** PUMP PARAMETRIC DATA ROUTINE *****

***** UNIVAC 1108 FORTRAN V *****

***** J.A.MCKAY 01949 B-201 X45178 *****

***** DATE CODED AUGUST 24, 1971 *****

***** REVISED MARCH 1972 *****

***** INPUT VARIABLES *****

IGS = 1 FOR OXYGEN
= 2 FOR HYDROGEN

JKM = 1 FOR MINIMUM POWER REQUIREMENT
= 2 FOR MINIMUM WEIGHT REQUIREMENT

DLP = REQUIRED PRESSURE RISE (PSI)

WDP = FLOWRATE REQUIRED (LG/SEC)

RNPS = NET POSITIVE SUCTION PRESSURE AVAILABLE (NPSA) (PSI)

RHW = FLUID DENSITY (LBS/CU.FT.)

***** OUTPUT VARIABLES *****

TOTNU = PUMP EFFICIENCY

V = PUMP VOLUME

E = PUMP POWER

WT = PUMP WEIGHT

PNSG = PUMP SPEED

NSTG = NUMBER OF PUMP STAGES

NPSPR = COMPUTED NPSH REQUIRED BY PUMP

SUBROUTINE PARPMP (IGS,JKM,DLP,WDP,RNPS,RHW,TOTNU,V,E,WT,PNSG,
I NSTG, NPSPR)

REAL NMAX,NPSA(2),NPSPR,NSSI,NUMX(5)

INCLUDE CIOUNT

INCLUDE CONST

INCLUDE DUMMY

INCLUDE SPUMP,LIST

INCLUDE TABLOK

DIMENSION DELP(2),RHO(2),S(2),WDOTP(2)

```

***** PARMP *****
58      DIMENSION AE(5),AV(5),AW(5)
59      DIMENSION XM(12),SWB(2)
60      C
61      EQUIVALENCE (SCRATCH(4),NUMX),(SCRATCH(12),DELP)
62      1,(SCRATCH(14),WDOTP),(SCRATCH(16),NPSPA),(SCRATCH(18),RHO)
63      2,(SCRATCH(34),AE),(SCRATCH(39),AV),(SCRATCH(44),AW)
64      C
65      DATA S/50000.,300000./
66      DATA XM /2.5,3.5,3.5,2.25,3.25,3.25,2.0,2.75,2.75,2.0,3.0,3.0 /
67      DATA SWB /50.,3./ PB /1000./
68      C
69      DELP(IGS) = DLP
70      NPSPA(IGS) = RNPS
71      RHO(IGS) = RHW
72      WDOTP(IGS) = WDP
73      DNS = 2.0
74      SGTQ = SQRT (449. * WDOTP(IGS) / RHO(IGS))
75      IF (NPSPA(IGS) .EQ. 0.) GO TO 10
76      C
77      NMAX = 5(IGS)*(144.*NPSPA(IGS)/RHO(IGS))**(.75) / SGTQ
78      IF (NMAX .LT. 100000.) GO TO 20
79      10 NMAX = 100000.
80      20 DO 50 I=1,5
81      C
82      C      I - NUMBER OF PUMP STAGES
83      C
84      NSTG = I
85      C
86      C      CALC HEAD RISE PER STAGE
87      C
88      H = 144.*DELP(IGS) / (I*RHO(IGS))
89      C
90      C      CALC MULT. FACTOR FOR SPECIFIC SPEED
91      C
92      XNS = SGTQ / H**(.75)
93      C
94      C      LOOP TO FIND MAX PUMP HYDRAULIC EFF. (NUMX)
95      C
96      DINT = NMAX
97      NSG = 0.
98      NSSI = 0.
99      DO 40 J=1,10
100     DINT = DINT / 2.
101     NSSI = NSG
102     NSG = NSG + DINT * 2.*DNS/XNS
103     CALL PUMPEF (2)
104     SD = DBLE(NU(2)) - DBLE(NU(1))
105     DINT = SIGN (DINT,SD)
106     NU(1) = NU(2)
107     IF (ABS(NSSI-NSS) - DNS) 42,42,40
108     40 CONTINUE
109     C
110     42 CONTINUE
111     NPSPR = 0.
112     IF (NPSPA(IGS) .EQ. 0.) GO TO 60
113     C
114     C      CALC. REQUIRED NPSP
115     C

```

```

***** PARMP *****
116      NPSPR = RHO(IGS) / 144.*(NSG*SQTG / S(IGS))**.4/.7.)
117      IF (NPSPA(IGS) .GE. NPSPR) GO TO 60
118      C
119      C      IF NPSP (REQ) GREATER THAN NPSP (AVIL) RECALC NU
120      C
121      NPSPR = NPSPA(IGS)
122      NSG = S(IGS) / SQTG *. (144.*NPSPR/RHO(IGS))**.3/.4.)
123      C
124      C      RECALC NSS,...,NU
125      C
126      DNS = 0.
127      CALL PUMPEF (1)
128      DNS = 2.0
129      60 CONTINUE
130      IF (U .GT. 1700.) NU = 0.
131      TOTNU = NU/(1+.05*(NSTG-1.)/NSTG)
132      ISTG = NSTG
133      IF (NSTG .GT. 3) ISTG = 3
134      IF (NSS .LE. 375.0) GO TO 120
135      C
136      C      FIND DIAMETER AND LENGTH OF PUMP
137      C      PUMP TYPE A OR C
138      C
139      PDI = 1.7 * DI
140      IF (DI .GT. 1.5) GO TO 100
141      PLGT = XM(ISTG)*DI
142      GO TO 130
143      100 IF (DI .GT. 2.0) GO TO 110
144      PLGT = XM(ISTG+3)*DI
145      GO TO 130
146      110 PLGT = XM(ISTG+6)*DI
147      GO TO 130
148      C
149      C      PUMP TYPE B
150      C
151      120 PDI = 1.4 * DI
152      PLGT = XM(ISTG+9)*DI
153      130 IF (NSTG .LT. 3) GO TO 140
154      PLGT = PLGT + DI*(NSTG-2) / 2.
155      140 CONTINUE
156      C
157      CALL FINTAB (NTBID(22))
158      WB = MIPE (1,DI)
159      FW = (WDOTP(IGS) / SWB(IGS))**.25)
160      P = NPSPA(IGS) + DELP(IGS)
161      C
162      C      CALCULATE PUMP WEIGHT
163      C
164      IF (NSS .GT. 375.0) GO TO 150
165      C
166      C      PUMP TYPE B
167      C
168      WI = .05
169      WH = .35
170      WS = NSTG**.75)
171      GO TO 160
172      C
173      C      PUMP TYPE A OR C

```

```

***** PARPMP *****
174 C
175 150 WI = .1*FW
176 WH = .55
177 WS = 1.
178 160 WI = WI*NSTG*WB
179 WH = WH*WB*NSTG**(.1/.3.)
180 WS = WS*.35*WB*P/PB*FW
181 WT = WI + WH + WS
182 C
183 C CALC. PUMP VOLUME
184 C
185 V = .25*PI*PLGT*PDI*PDI
186 C
187 C CALC. PUMP POWER
188 C
189 E = WDOTP(IGS)*H*NSTG / (550.*TOTNU)
190 AE(I) = E
191 AV(I) = V
192 AW(I) = WT
193 NUMX(I) = TOTNU
194 C
195 50 CONTINUE
196 XMN = 1.E10
197 DO 230 K=1,5
198 GO TO (200,210),JKM
199 C
200 C FOR MIN. POWER REQ.
201 C
202 200 IF (AE(K) .GE. XMN) GO TO 230
203 XMN = AE(K)
204 GO TO 220
205 C
206 C FOR MIN. WEIGHT REQ.
207 C
208 210 IF (AW(K) .GE. XMN) GO TO 230
209 XMN = AW(K)
210 220 NSTG = K
211 230 CONTINUE
212 E = AE(NSTG)
213 V = AV(NSTG)
214 WT = AW(NSTG)
215 TOTNU = NUMX(NSTG)
216 PMSG = MSG
217 C
218 C OFF-DESIGN PUMP PERFORMANCE
219 C
220 RETURN
221 END

```

***** SUBROUTINE PFND

```

1      SUBROUTINE PFND(T,D,P)
2      COMMON /RFPR/ RF(10)
3      COMMON /CEOS/G(41)
4      COMMON /SCRH/ B(40)
5
6      C
7      C.... THIS ROUTINE CALCULATES PRESSURE GIVEN TEMPERATURE AND DENSITY
8      C.... FROM THE EQUATION OF STATE
9      C
10     R=RF(5)
11     D2=D*D
12     D3=D2*D
13     D4=D3*D
14     D5=D4*D
15     D6=D5*D
16     D7=D6*D
17     D8=D7*D
18     D9=D8*D
19     D10=D9*D
20     D11=D10*D
21     D12=D11*D
22     TS = SQRT(T)
23     T2=T*T
24     T3=T2*T
25     T4=T3*T
26     GM=G(41)
27     F = EXP(GM*D2)
28     B( 1)=D2*T
29     B( 2)=D2*TS
30     B( 3)=D2
31     B( 4)=D2/T
32     B( 5)=D2/T2
33     B( 6)=D3*T
34     B( 7)=D3
35     B( 8)=D3/T
36     B( 9)=D3/T2
37     B(10)=D4*T
38     B(11)=D4
39     B(12)=D4/T
40     B(13)=D5
41     B(14)=D6/T
42     B(15)=D6/T2
43     B(16)=D7/T
44     B(17)=D8/T
45     B(18)=D8/T2
46     B(19)=D9/T2
47     B(20)=D3*F/T2
48     B(21)=D3*F/T3
49     B(22)=D5*F/T2
50     B(23)=D5*F/T4
51     B(24)=D7*F/T2
52     B(25)=D7*F/T3
53     B(26)=D9*F/T2
54     B(27)=D9*F/T4
55     B(28)=D11*F/T3
56     B(29)=D11*F/T3
57     B(30)=D13*F/T2

```



```

*****      PFND      :*****
58          B(31)=D13*F/T3
59          B(32)=D13*F/T4
60          N=32
61          P = 0.0
62          DO 1 I=1,N
63      1    P=P+B(I)*G(I)
64          P=P+R*D*T
65          RETURN
66          END

```

***** SUBROUTINE PFNDB

```
1      SUBROUTINE PFNDB(TB,DB,PB)
2      COMMON /RFPR/ RF(10)
3      WT=RF(7)
4      T = TB/1.8
5      D = DB * 453.59237E-3/(WT * 2.8916847E-2)
6      CALL PFND(T,D,P)
7      PB = P * 1.01325E+5/6.8947572E+3
8      RETURN
9      END
```

***** SUBROUTINE PHIB

```
1      SUBROUTINE PHIB(DB,CVB,D2B,EDB)
2      C
3      C   CALCULATES THE ENERGY DERIVATIVE FROM THE EQUATION OF STATE IN
4      C   BRITISH UNITS
5      C   MUST FOLLOW CALLS OF PROPB,CPVTDB,AND DPDTB TO DEFINE DB, CVB, &
6      C   D2
7      C   INPUT AND OUTPUT PARAMETERS ARE IN BRITISH UNITS
8      C
9      VB = 1.0/DB
10     EDB=(VB/CVB)*D2B
11     RETURN
12     END
```

***** SUBROUTINE PHTHON

```

1      SUBROUTINE PHTHON(TEMP,DENS,NGAS,PHI,THETA)
2      C
3      COMMON /METH/ M
4      C
5      C      INPUT TO THIS SUBROUTINE MUST BE IN BRITISH UNITS
6      C
7      T = TEMP
8      D = DENS
9      M = 1
10     C
11     IF(NGAS.EQ.1) KF = 1
12     IF(NGAS.EQ.18) KF = 2
13     C
14     C      KF = 1 CALL IN OXYGEN PARAMETERS
15     C      KF = 2 CALL IN NITROGEN PARAMETERS
16     C
17     IF(KF.EQ.1) CALL DATA02
18     IF(KF.EQ.2) CALL DATAN2
19     C
20     D1=DPDDB(T,D)
21     D2=DPDTB(T,D)
22     CALL CPVTDB(T,D,CP,CV)
23     CALL THETAB(D,CP,D1,D2,SHI)
24     THETA = SHI
25     C
26     CALL PHIB(D,CV,D2,ED)
27     PHI = ED
28     C
29     RETURN
30     END

```

SUBROUTINE PROP

```

1      SUBROUTINE PROP(T,P,D,K,H,S,U,Z)
2      COMMON /CRPR/ CR(3) /METH/M
3
4      C.... GENERALIZED PROPERTY CALCULATOR
5
6      C.... ROUTINE CALCULATES PROPERTIES FOR FOLLOWING INPUT OF K
7
8      C      K=1  INPUT IS T + P  RETURNS D, H, S, + U
9      C      K=2  INPUT IS T + D  RETURNS P, H, S, + U
10     C      K=3  INPUT IS T      RETURNS P, D, H, S, + U FOR SATURATED VAPOR
11     C      K=4  INPUT IS T      RETURNS P, D, H, S, + U FOR SATURATED LIQUID
12
13     C.... NOTE      ALL REAL VARIABLES IN CALL STATEMENTS TO ROUTINES IN
14     C               THIS PACKAGE MUST BE TYPED IMPLICIT DOUBLE PRECISION
15     C               (A-H, O-Z)
16
17     C.... NOTE      THE FIRST CALL STATEMENT IN THE USER'S MAIN PROGRAM MUST
18     C               BE TO A DATA INITIALIZATION ROUTINE
19     C               EXAMPLE  CALL DATA02
20
21     C.... NOTE      THE METHOD OF PROPERTY CALCULATION IS DETERMINED BY THE
22     C               VALUE OF M CONTAINED IN COMMON BLOCK /METH/M
23
24     C      M=1  INDICATES PROPERTY CALCULATION TO BE CARRIED OUT BY
25     C            CONTINUOUS INTEGRATION OF ISOTHERMS THROUGH THE TWO PHASE
26     C            REGION
27     C      M=2  INDICATES PROPERTY CALCULATION IS INTERRUPTED AT THE TWO
28     C            PHASE VAPOR BOUNDARY AND THE CLAPEYRON RELATION WITH THE
29     C            VAPOR PRESSURE EQUATION IS USED TO CALCULATE THE LATENT
30     C            HEAT. INTEGRATION OF ISOTHERMS IS CONTINUED AT THE
31     C            SATURATED LIQUID BOUNDARY
32
33     PC=CR(1)
34     DC=CR(2)
35     TC=CR(3)
36     IF((K.GT.0).AND.(K.LT.5))GO TO 1
37     WRITE(6,300)K
38 300 FORMAT(' *** ERROR IN CALL PROP ***',/,
39           '      K MUST EQUAL 1,2,3, OR 4',/,
40           '      K = ',I10)
41     RETURN
42     1 IF(K.LT.3)GO TO 3
43     IF(T.LE.TC)GO TO 2
44     WRITE(6,301)T
45 301 FORMAT(' *** ERROR IN CALL PROP ***',/,
46           '      SATURATION PROPERTIES HAVE BEEN REQUESTED',/,
47           '      FOR A TEMPERATURE THAT EXCEEDS CRITICAL',/,
48           '      T = ',G15.5)
49     2 P=VPN(T)
50     IF(K.EQ.3)CALL DFND(T,P,D,ZI,2)
51     IF(K.EQ.4)CALL DFND(T,P,D,ZI,1)
52     GO TO 4
53     3 IF(K.GT.1)GO TO 7
54     CALL DFND(T,P,D,ZI,0)
55     IF(T.GT.TC)GO TO 5
56     4 IF(D.GT.DC)GO TO 6
57     5 CALL VPROP(T,P,D,3,H,S,U,Z)

```

```

*****  PROP  *****
58      RETURN
59      6 IF(M.EQ.1)GO TO 5
60      CALL LPROP(T,P,D,3,H,S,U,Z)
61      RETURN
62      7 IF(T.GT.TC)GO TO 8
63      VP=VPN(T)
64      CALL DFND(T,VP,DV,Z2,2)
65      CALL DFND(T,VP,DL,Z2,1)
66      IF(D.GE.DL)GO TO 9
67      IF(D.GT.DV)GO TO 10
68      8 CALL VPROP(T,P,D,1,H,S,U,Z)
69      RETURN
70      9 IF(M.EQ.1)GO TO 8
71      CALL LPROP(T,P,D,1,H,S,U,Z)
72      RETURN
73      10 VL = 1.0/DL
74      VV = 1.0/DV
75      V = 1.0/D
76      X=(V-VL)/(VV-VL)
77      CALL VPROP(T,P,DV,1,HV,SV,UV,ZV)
78      IF(M.EQ.2)GO TO 11
79      CALL VPROP(T,P,DL,1,HL,SL,UL,ZL)
80      GO TO 12
81      11 CALL LPROP(T,P,DL,1,HL,SL,UL,ZL)
82      12 H=HL+X*(HV-HL)
83      S=SL+X*(SV-SL)
84      U=UL+X*(UV-UL)
85      Z = ZL
86      RETURN
87      END

```

***** SUBROUTINE PRØPB

```
1      SUBROUTINE PROPB(TB,PB,DB,K,HB,SB,UB,ZB)
2      COMMON /RFPR/RF(10)
3      NT=RF(7)
4      T = TB/1.8
5      P = PB * 6.8947572E+3/(1.01325E+5
6      D = DB * 453.59237E-3/(WT * 2.8316847E-2)
7      CALL PROP(T,P,D,K,H,S,U,Z)
8      PB = P * 1.01325E+5/6.8947572E+3
9      DB = D * WT * 2.8316847E-2/453.59237E-3
10     HB = H * 453.59237/(1.0543502E+3 * WT)
11     UB = U * 453.59237/(1.0543502E+3 * WT)
12     SB = S * 453.59237/(1.0543502E+3 * 1.8 * WT)
13     ZB = Z
14     RETURN
15     END
```

FUNCTION PSATH

1	FUNCTION PSATH(PRESS,HG,HL)	1821	0001
2	DIMENSION R(19),TL(19),TG(19),TF(19)	1822	0002
3	DATA R/1.022,2.0,4.0,8.0,14.0,25.0,43.0,69.0,99.0,128.0,151.0,	1823	0003
4	1165.,176.0,182.0,185.0,186.5,187.25,187.46875,187.506/	1824	0004
5	DATA TF/24.845,	1825	0005
6	227.07,29.81,33.07,36.18,39.96,44.12,48.33,51.97,54.79,56.72,57.80,	1826	0006
7	358.57,58.99,59.18,59.29,59.34,59.353,59.356/	1827	0007
8	DATA TG/60.31,65.11,70.59	1828	0008
9	4,76.35,80.98,85.11,87.40,86.54,81.94,74.15,64.83,56.86,47.34,39.56	1829	0009
10	5,33.46,28.34,22.31,18.66,16.55/	1830	0010
11	DATA TL/-132.8,-129.13,-124.25,-117.79	1831	0011
12	6,-110.86,-101.3,-89.04,-74.22,-58.58,-43.43,-30.07,-20.56,-11.13,	1832	0012
13	7-4.27,1.17,5.54,10.83,14.29,16.36/	1833	0013
14	P=PRESS	1834	0014
15	IF(P.LT.1.022)P=1.022	1835	0015
16	IF(P.GE.187.506)P=187.506	1836	0016
17	DO 104 I=2,19	1837	0017
18	IF(P-R(I))102,101,104	1838	0018
19	101 HL=TL(I)	1839	0019
20	HG=TG(I)	1840	0020
21	PSATH =TF(I)	1841	0021
22	RETURN	1842	0022
23	102 D=R(I)-R(I-1)	1843	0023
24	PPR=R(I)-P	1844	0024
25	PPR=P-R(I-1)	1845	0025
26	HL=(TL(I)*PPR+TL(I-1)*PPR)/D	1846	0026
27	HG=(TG(I)*PPR+TG(I-1)*PPR)/D	1847	0027
28	PSATH =(TF(I)*PPR+TF(I-1)*PPR)/D	1848	0028
29	RETURN	1849	0029
30	104 CONTINUE	1850	0030
31	RETURN	1851	0031
32	END	1852	0032

***** PTDENS *****

```

58 DATA AD /5.033,5.112,5.183,4.816,4.910,4.994,4.558,4.
59 1676,4.764,4.248,4.405,4.533,3.880,4.094,4.246,3.442,3.743,3.958,2.
60 2953,3.361,3.633,2.481,2.975,3.307,2.093,2.616,3.002,1.804,2.309,2.
61 3698,1.588,2.059,2.462,1.423,1.856,2.226,1.293,1.691,2.056,1.187,1.
62 4555,1.886,1.100,1.441,1.754,1.026,1.344,1.639,1.962,1.262,1.546,1.9
63 5076,1.189,1.453,4.742,4.853,4.946,5.031,4.430,4.584,4.709,4.816
64 6,4.024,4.279,4.417,4.558,3.165,3.778,4.050,4.248,1.430,2.988,3.595
65 7,3.880,1.023,3.558,1.189,1.804,1.015,1.503,1.05,1.588,1.009,1.
66 84591,1.945,1.423,1.0065,1.423,1.8618,1.293,1.0042,1.3928,1.7943,1.187,
67 9,1.0025,1.3669,1.7379,1.1,1.0015,1.3445,1.6898,1.026,1.0007,1.3249/
68 DATA AE /1.6483,1.9627,1.0001,1.3075,1.6120,1.9076/
69 DATA AF /1.07704,1.02568,1.1307,1.2384,1.06786,1.02262,1.
70 1144,1.2079,1.06066,1.02022,1.018,1.1844,1.05484,1.01828,1.09179,1.1659,1.
71 2,05004,1.01668,1.0836,1.1508,1.04602,1.01534,1.07677,1.1383,1.3887,1.9
72 386,4.105,3.447,3.646,3.792,2.910,3.226,3.442,2.323,2.760,3.051,1.8
73 457,2.317,2.670,1.549,1.963,2.326,1.34,1.701,2.035,1.189,1.497,1.80
74 54,1.7859,1.309,1.857,1.6864,1.106,1.549,1.614,1.9689,1.34,1.5581,1.8
75 6735,1.189,1.1675,1.3524,1.5577,1.7859,1.1530,1.3178,1.4955,1.6864,1.409,
76 7,2902,1.4479,1.6141,1.1307,1.2674,1.4127,1.5581,1.03228,1.03224,1.09858,
77 8,1675,1.029742,1.02971,1.09043,1.153,1.027561,1.02754,1.08359,1.1409,1.0
78 925680,1.02568,1.07819,1.1307/
79 DATA AG /1.226,1.807,2.329,2.674,2.91,1.052,1.481,1.9
80 141,2.326,2.616,1.9378,1.279,1.656,2.019,2.323,1.8526,1.141,1.456,1.7
81 274,2.089,1.7859,1.047,1.309,1.583,1.857,1.09975,1.2084,1.3279,1.461,1.6
82 3115,1.7847,1.9864,1.226,1.09478,1.1962,1.3063,1.4264,1.5585,1.7052,1.8693,
83 41,052,1.08981,1.1855,1.2878,1.3977,1.5164,1.6451,1.7851,1.9388,1.08590,1.176
84 5,2717,1.3734,1.4817,1.5972,1.7207,1.8526,1.08199,1.1675,1.2599,1.3524,1.455
85 60,1.5577,1.6718,1.7859,1.019364,1.01932,1.05888,1.09975,1.017558,1.0175
86 73,1.05323,1.08981,1.01615,1.01611,1.04882,1.08199,1.304,1.422,1.541,
87 83,659,1.777,1.3044,1.3290,1.450,1.572,1.667,2.740,1.102,1.304,1.449,
88 93,557,2.322,2.880,1.141,1.315,1.3447,1.861,2.621,2.959,1.170,1.313/
89 DATA AH /1.551,2.329,2.760,1.014,3.179,1.358,2.045,
90 12,546,2.848,1.044,1.226,1.807,2.329,2.674,2.910/
91 DATA AI /1.6295,2.42,2.726,2.849,2.937,3.005,3.061,3.1
92 11,3.153,3.191,3.227,3.259,1.9338,1.076,1.265,1.599,2.202,2.506,2.65
93 29,2.768,2.85,2.918,2.976,3.026,1.8297,1.9234,1.03,1.159,1.322,1.542,
94 31,835,2.136,2.356,2.509,2.618,2.709,1.7590,1.8323,1.9121,1.001,1.103,
95 41,220,1.359,1.523,1.713,1.916,2.106,2.267,1.7053,1.7670,1.8326,1.9039,
96 5,9807,1.066,1.160,1.265,1.383,1.513,1.655,1.803,1.6621,1.7161,1.7728,
97 6,8331,1.8976,1.9656,1.040,1.120,1.206,1.299,1.400,1.507,1.6259,1.6745,
98 7,250,1.7780,1.8337,1.8927,1.9537,1.019,1.089,1.163,1.242,1.325,1.5941,
99 8,6412,1.6883,1.7355,1.7826,1.8372,1.8921,1.9469,1.005,1.070,1.135,1.200/
100 DATA AJ /1.000743,1.04849,1.09854,1.0,1.000787,1.04023,1.
101 108466,1.1315,1.1915,1.000437,1.03393,1.07032,1.1097,1.1534,1.000257,1.029
102 241,1.06027,1.09278,1.1275,1.000177,1.02599,1.05289,1.0808,1.1098,1.000122
103 3,1.0233,1.0472,1.07177,1.09714,1.000083,1.02112,1.04266,1.06465,1.08719,1.
104 4000065,1.01932,1.03810,1.05888,1.07931,1.0,1.01172,1.02363,1.000111,1.00
105 59736,1.0198,1.000059,1.008154,1.01648,1.000036,1.007021,1.01414,1.00002
106 64,1.006166,1.0124,1.000015,1.005498,1.01104,1.000011,1.004962,1.009954,1.
107 7,000009,1.004521,1.009063,1.000007,1.004153,1.008321,1.000022,1.003846,
108 8,007715/
109 DATA AK /1.664,1.248,0.0,0.0,0.0,1.469,1.2418,1.3397,0.0,0.0,0.0,1.315,
110 1,2117,1.3091,1.4233,1.5318,0.0,1.195,1.1891,1.2684,1.3626,1.4811,1.6115,
111 2,1098,1.1717,1.2400,1.3166,1.4051,1.5124,1.1017,1.1578,1.2183,1.2841,1.3568,
112 3,4387,1.09486,1.1463,1.2009,1.2592,1.3221,1.3903,1.08892,1.1377,1.1865,1.2407
113 4,295,1.3565,1.5712,1.727,1.9725,0.0,0.0,1.5124,1.6559,1.8514,1.116,0.0,
114 50,1.4755,1.5827,1.7329,1.9604,1.252,1.718,1.4387,1.5333,1.6477,1.7994,1.0
115 668,1.534,1.4145,1.4956,1.5914,1.7060,1.8535,1.082,1.3903,1.4653,1.5491,1.64

```

***** PTDENS *****

```

116 747,.7575,.8985,.3734,.4399,.5151,.5986,.6928,.8020,.9565,.4181,.48
117 898,.5616,.6487,.7358,.08892,.1865,.295,.4181,.5616,.7358,.08373,.1
118 9743,.2731,.3822,.5044,.6439,.07931,.1649,.2562,.3545,.4610,.5814/
119 P=PRES
120 IF(P,LT,1.0) P=1.0
121 T=TEMP
122 IF(T,LT,180.0) GO TO 8
123 IF(T,GE,1300.0) GO TO 4
124 IF(T,GE,490.0) GO TO 2
125 IF(P,GE,800.) GO TO 1
126 N=1
127 GO TO 33
128 1 N=2
129 GO TO 33
130 2 IF(P,GE,300.0) GO TO 3
131 N=3
132 GO TO 33
133 3 N=4
134 GO TO 33
135 4 IF(T,GE,2500.0) GO TO 6
136 IF(P,GE,100.0) GO TO 5
137 N=5
138 GO TO 33
139 5 N=6
140 GO TO 33
141 6 IF(T,GE,5000.0) T=4999.99999
142 IF(P,GE,10.0) GO TO 7
143 N=7
144 GO TO 33
145 7 N=8
146 GO TO 33
147 8 TZ=24.84+0.00317*P
148 IF(T,LT,TZ) T=TZ
149 IF(P,LT,881.76) GO TO 11
150 IF(P,LT,2645.28) GO TO 9
151 N=9
152 GO TO 33
153 9 IF(P,LT,1469.6) GO TO 10
154 N=10
155 GO TO 33
156 10 N=11
157 GO TO 33
158 11 IF(T,GE,59.4) GO TO 14
159 N=12
160 IF(P,GE,187.6385) GO TO 33
161 DO 12 I=2,20
162 IF(P-PS(I))13,13,12
163 12 CONTINUE
164 I=20
165 13 TM=TS(I-1)+(TS(I)-TS(I-1))*(P-PS(I-1))/(PS(I)-PS(I-1))
166 IF(T,GE,TM) GO TO 24
167 GO TO 33
168 14 IF(T,LT,108.0) GO TO 16
169 IF(P,LT,132.264) GO TO 15
170 N=13
171 GO TO 33
172 15 N=14
173 GO TO 33

```

```

***** PTDENS *****
174 16 IF(P.LT.587.84) GO TO 17
175 N=15
176 GO TO 33
177 17 IF(T.LT.72.0) GO TO 23
178 IF(T.LT.86.4) GO TO 20
179 IF(P.LT.293.92) GO TO 18
180 N=16
181 GO TO 33
182 18 IF(P.LT.73.48) GO TO 19
183 N=17
184 GO TO 33
185 19 N=18
186 GO TO 33
187 20 IF(P.LT.293.92) GO TO 21
188 N=19
189 GO TO 33
190 21 IF(P.LT.36.74) GO TO 22
191 N=20
192 GO TO 33
193 22 N=21
194 GO TO 33
195 23 IF(P.LT.293.92) GO TO 24
196 N=22
197 GO TO 33
198 24 IF(P.LT.180.0) GO TO 25
199 N=23
200 GO TO 33
201 25 IF(P.GE.29.0) GO TO 27
202 IF(P.LT.2.9392) GO TO 26
203 N=24
204 GO TO 33
205 26 N=25
206 GO TO 33
207 27 IF(T.GE.64.8) GO TO 29
208 IF(P.GE.102.0) GO TO 28
209 N=26
210 GO TO 33
211 28 N=27
212 GO TO 33
213 29 N=28
214 33 FP=(P-BP(N))/DP(N)
215 IP=FP
216 IF(IP.GT.MX(N)) IP=MX(N)
217 FI=IP
218 F=FP-FI
219 FP=1.0-F
220 FT=(T-BT(N))/DT(N)
221 IT=FT
222 FI=IT
223 FF=FT-FI
224 FT=1.0-FF
225 I=IT*JP(N)+IP*LOC(N)
226 J=I+JP(N)
227 PTDENS=FP*FT*R(I)+P*FT*R(I+1)+FP*FF*R(J)+P*FF*R(J+1)
228 RETURN
229 END

```

FUNCTION PHEAT

1	FUNCTION PHEAT(PRES,TEMP,KTRANS)	1303 0001
2	COMMON/SPHEAT/CP(823),CV(823)	1304 0002
3	DIMENSION LOC(23),JP(19),HX(19),BP(19),DP(19),BT(23),DT(19),PS(12)	1305 0003
4	1,TS(12),TL(10)	1306 0004
5	DATA PS/1.022,2.4,8.14,25.43,69.99,128.151,160./	1307 0005
6	DATA TS/24.845,27.07,29.81,33.07,36.18,39.96,44.12,48.33,51.97,	1308 0006
7	154.79,56.72,57.46/	1309 0007
8	DATA LOC/1.50,71,113,133,141,170,165,228,264,348,372,426,492,528,	1310 0008
9	1584,626,638,734,654,682,694,718/	1311 0009
10	DATA JP/7,3,6,5,2,4,9,9,4,7,4,9,6,6,7,7,2,5/	1312 0010
11	DATA HX/5,1,4,3,0,2,2,4,2,5,2,7,4,4,5,5,0,0,3/	1313 0011
12	DATA BP/0,20,0,1000,100,1000,1469.6,0,1469.6,587.84,987.84	1314 0012
13	1,0,100,100,40,40,0,0,1./	1315 0013
14	DATA BT/2600,2600,2600,2600,800,300,120,120,25,2,27,72,.	1316 0014
15	181,25,56,26,41,25,25,2600,5000,5000,5000,5000./	1317 0015
16	DATA DP/5,40,200,1000,4900,2000,1175.68,293.92,1175.68,	1318 0016
17	1146.96,146.96,73.48,100,100,20,20,40,40,1./	1319 0017
18	DATA DT/400,400,400,800,600,100,30,30,12.6,9,3.6,9,5.,	1320 0018
19	15.4,8,4,8,200./	1321 0019
20	DATA TL/24.846,27.175,29.310,31.299,33.176,34.962,36.672,38.317,	1322 0020
21	139.904,41.456/	1323 0021
22	1 P=PRES	1324 0022
23	IF(P.LT.1.0) P=1.0	1325 0023
24	T=TEMP	1326 0024
25	KTR=KTRANS	1327 0025
26	IF(T.LT.126.) GO TO 9	1328 0026
27	IF(T.LT.2600.) GO TO 5	1329 0027
28	IF(T.GE.6000.) T=5999.99999	1330 0028
29	IF(P.GE.100.) GO TO 3	1331 0029
30	IF(P.GE.30.) GO TO 2	1332 0030
31	IF(P.GE.5.) GO TO 20	1333 0031
32	N=19	1334 0032
33	GO TO 33	1335 0033
34	GO TO 33	1336 0034
35	20 N=1	1337 0035
36	N=20	1338 0036
37	GO TO 33	1339 0037
38	2 N=2	1340 0038
39	N=21	1341 0039
40	GO TO 33	1342 0040
41	3 IF(P.GE.1000.) GO TO 4	1343 0041
42	N=3	1344 0042
43	N=22	1345 0043
44	GO TO 33	1346 0044
45	4 N=4	1347 0045
46	N=23	1348 0046
47	GO TO 33	1349 0047
48	5 IF(T.LT.300.) GO TO 7	1350 0048
49	IF(T.LT.800.) GO TO 6	1351 0049
50	N=5	1352 0050
51	GO TO 33	1353 0051
52	6 N=6	1354 0052
53	GO TO 33	1355 0053
54	7 IF(P.LT.1469.6) GO TO 8	1356 0054
55	N=7	1357 0055
56	GO TO 33	1358 0056
57	8 N=8	1359 0057

***** PTHEAT *****

58	GO TO 33	1360 0058
59	9 IF(P.LT.587.84) GO TO 12	1361 0059
60	IF(P.LT.1469.6) GO TO 10	1362 0060
61	N=9	1363 0061
62	GO TO 30	1364 0062
63	10 IF(P.LT.1028.72.AND.T.GE.72.0.AND.T.LT.90.0) GO TO 11	1365 0063
64	N=10	1366 0064
65	GO TO 30	1367 0065
66	11 N=11	1368 0066
67	GO TO 33	1369 0067
68	12 IF(T.LT.81.) GO TO 13	1370 0068
69	N=12	1371 0069
70	GO TO 33	1372 0070
71	13 IF(P.LT.160.) GO TO 15	1373 0071
72	TM=((1.86867647E-7*P-.12613701E-3)*P+.10353383)*P+.43.8056878	1374 0072
73	IF(T.GT.TM) GO TO 14	1375 0073
74	N=13	1376 0074
75	GO TO 30	1377 0075
76	14 N=14	1378 0076
77	GO TO 33	1379 0077
78	15 DO 16 I=2,12	1380 0078
79	IF(P-PS(I))17,17,16	1381 0079
80	16 CONTINUE	1382 0080
81	I=12	1383 0081
82	17 TM=TS(I-1)+(TS(I)-TS(I-1))*(P-PS(I-1))/(PS(I)-PS(I-1))	1384 0082
83	IF(T.GE.TM) GO TO 18	1385 0083
84	N=15	1386 0084
85	IF(P.LT.40.) N=17	1387 0085
86	GO TO 30	1388 0086
87	18 N=16	1389 0087
88	IF(P.LT.40.) N=18	1390 0088
89	GO TO 33	1391 0089
90	30 F=P/587.84	1392 0090
91	I=F	1393 0091
92	IF(I.GT.8) I=8	1394 0092
93	FI=I	1395 0093
94	F=F-FI	1396 0094
95	TQ=(1.0-F)*TL(I+1)+F*TL(I+2)	1397 0095
96	IF(T.LT.TQ) T=TQ	1398 0096
97	33 IF(T.LE.5000.)NI=N	1399 0097
98	FP=(P-AP(N))/DP(N)	1400 0098
99	IP=FP	1401 0099
100	IF(IP.GT.MX(N)) IP=MX(N)	1402 0100
101	FI=IP	1403 0101
102	F=FP-FI	1404 0102
103	FP=1.0-F	1405 0103
104	FT=(T-BT(N1))/DT(N)	1406 0104
105	IT=FT	1407 0105
106	FI=IT	1408 0106
107	FF=FT-FI	1409 0107
108	FT=1.0-FF	1410 0108
109	I=IT*JP(N)+IP*LOC(N1)	1411 0109
110	J=I+JP(N)	1412 0110
111	IF(KTR.EQ.2) GOTO 37	1413 0111
112	CTCP=FP*FT*CP(I)+F*FT*CP(I+1)+FP*FF*CP(J)+F*FF*CP(J+1)	1414 0112
113	IF(N.LT.13.OR.N.GE.17) GO TO 36	1415 0113
114	IF(N.LT.15) GO TO 35	1416 0114
115	CTCP=CTCP/(187.506-P*ABS(T-TM)*28.13)	1417 0115

***** PTHEAT *****

116	GO TO 36	1418 0116
117	35 CTCP=CTCP/(ABS (T-TM)/1.8+ABS (P-187.506)*.008008982)	1419 0117
118	36 IF(KTR.GE.2) GO TO 37	1420 0118
119	PTHEAT=CTCP	1421 0119
120	RETURN	1422 0120
121	37 PTHEAT=FP*FT*CV(I)+F*FT*CV(I+1)+FP*FF*CV(J)+F*FF*CV(J+1)	1423 0121
122	IF(KTR.LT.3) RETURN	1424 0122
123	PTHEAT=CTCP/PTHEAT	1425 0123
124	RETURN	1426 0124
125	END	1427 0125

***** SUBROUTINE PVAPOR

```

1      SUBROUTINE PVAPOR(T,I,P)
2      GO TO (1,2,3,2,5,6,7,6,7,10,3,12,13,14,15,16),I
3      1 P=EXP(12.04-1519./T)
4      RETURN
5      2 P=10.**(2.9303-79.821/T+.011628*T)
6      RETURN
7      3 P = EXP(11.63-1374./T)
8      IF(P.GT.200.) P = EXP(13.43-1763./T)
9      RETURN
10     5 P = .825*EXP(11.63-1374./T)+.175*EXP(12.04-1519./T)
11     IF(P.GT.200.) P = .825*EXP(13.43-1763./T)+.175*EXP(12.04-1519./T)
12     RETURN
13     6 P = EXP(11.83-1839./T)
14     RETURN
15     7 P = 10.**(5.73-1050./T)
16     RETURN
17     10 P = EXP(12.3579-3168.7/T)
18     RETURN
19     12 P = EXP(14.45-5090./T)
20     RETURN
21     13 P = EXP(16.54098-7.3483*(1000./T))
22     RETURN
23     14 P = EXP(13.4055-6.65*(1000./T))
24     RETURN
25     15 PLOGMM = 7.4837-1.8*1197./T
26     P = .01934*(10.**PLOGMM)
27     RETURN
28     16 PLOGMM = 8.2875-1.8*1996./T
29     P = .01934*(10.**PLOGMM)
30     RETURN
31     END

```

SUBROUTINE RHOLIQ

```

1      SUBROUTINE RHOLIQ(T,I,R)
2      GO TO (1,2,3,2,5,6,7,6,7,10,3,12,13,14,15,16),I
3      1 R = 62.43*(1.6983-.003418*T)
4      RETURN
5      2 R=18.25132T-1.4162715*T+.058929936*T*T-.00121374846*T*T*T+
6      1 .121340376E-4*T**4-.4834473E-7*T**5
7      RETURN
8      3 R = 135.8864-.272046*T
9      RETURN
10     5 R = 129.5-.25*T
11     RETURN
12     6 R = 38.38-.0589*T
13     RETURN
14     7 R = 134.45-.169*T
15     RETURN
16     10 R = 62.43*(.76188-.000978*T)
17     RETURN
18     12 R = 62.4153-.046113*T
19     RETURN
20     13 R = 131.336-.07795*T
21     RETURN
22     14 R = 73.374-.032187*T
23     RETURN
24     15 R = 62.43*(2.805909-.001944*T)
25     RETURN
26     16 R = 62.43*(1.25214-.000453*T)
27     RETURN
28     END

```


***** SUBROUTINE SPHSEG

```

1  C
2  SUBROUTINE SPHSEG (PVOL,RAD,H)
3  C
4  INCLUDE CONST,LIST
5  C
6  DIMENSION Y(3)
7  C
8  C      CALC. VOL. OF TOTAL HEMISPHERE
9  TVOL = PI203 * RAD**3
10 GO TO 70
11 C      ENTRY FOR ELLIPTICAL SPHEROID
12 C      RAD ALONG AXIS OF ROTATION
13 ENTRY ELIPSG (PVOL,RAD,RPD,H)
14 C
15 TVOL = PI203*RPD*RPD*RAD
16 70 IF (PVOL .NE. 0.) GO TO 80
17 H = 0.
18 RETURN
19 80 CONTINUE
20 XM = PVOL / TVOL
21 IF (XM .GT. 0.) GO TO 90
22 WRITE (6,1002) XM
23 RETURN
24 90 CONTINUE
25 PHI3 = ACOS (1.0-XM) / 3.0
26 DO 100 I=1,3
27 XI = I - 1
28 Y(I) = RAD*(1.0 + 2.0*COS (PHI3 + XI*PI203))
29 100 CONTINUE
30 DO 200 I=1,3
31 K = I
32 IF (Y(I) .GT. 0. .AND. Y(I) .LT. RAD) GO TO 120
33 200 CONTINUE
34 WRITE (6,1000) Y
35 RETURN
36 120 H = Y(K)
37 RETURN
38 C
39 C      FIND HEAD IN CYLINDER
40 C
41 ENTRY CYLHED (PVOL,RAD,H)
42 H = PVOL / (PI*RAD*RAD)
43 RETURN
44 C
45 C      CALC. HEAD IN FRUSTRUM OF CONE
46 C
47 ENTRY FRHEAD (PVOL,RTOP,RBOT,HGT,H)
48 RBMT = RBOT - RTOP
49 VD = (PI/3.0)*HGT/RBMT*RBOT*RBOT*RBOT - PVOL
50 H = (PI*RBOT*HGT-(3.0*(PI*HGT)**2*RBMT*VD)**(1./3.)) /
51 (PI*RBMT)
52 RETURN
53 C
54 C      RAD ALONG AXIS OF ROTATION
55 ENTRY CYMSPH (PVOL,RAD,RPD,H)
56 C
57 TO CALC. HEAD IN A VOLUME BETWEEN A CYLINDER AND

```

***** SPHSEG *****

```
58 C          SPHEROID
59   D = 3.0*RAD*RAD*PVOL / (PI*RPD*RPD)
60   H = RAD - (RAD**3 - D)**(1./3.)
61   RETURN
62 C
63   1000 FORMAT ('D' 10X 'COULD NOT FIND H FOR HEMISPHERE' 3F(5.4))
64   1002 FORMAT ('D' 10X 'ERROR INPUT TO SPHSEG' F(5.7))
65 C
66   END
```

BLØCK DATA - SPHTDA

1	BLOCK DATA	1429	0001
2	COMMON/SPHEAT/AA(111),AB(111),AC(111),AD(111),AE(116),AF(112),	1430	0002
3	AG(111),AH(40),	1431	0003
4	2 AJ(110),AJ(111),AK(111),AL(111),AM(116),AN(112),AO(111)	1432	0004
5	3,AP(41)	1433	0005
6	DATAAA/3.804,3.796,3.794,3.794,3.793,3.793,3.793,4.025,3.951,3.937	1434	0006
7	1,3.931,3.927,3.924,3.922,4.655,4.259,4.184,4.15,4.13,4.117,4.107,6	1435	0007
8	2,495,5.032,4.755,4.632,4.559,4.509,4.472,10.98,6.839,6.054,5.706,5	1436	0008
9	3,498,5.356,5.252,20.06,10.48,8.646,7.834,7.35,7.02,6.776,35.35,16.	1437	0009
10	483,13.18,11.55,10.58,9.919,9.429,3.793,3.792,3.792,3.924,3.917,3.9	1438	0010
11	513,4.117,4.076,4.059,4.51,4.359,4.298,5.36,4.931,4.757,7.028,6.027	1439	0011
12	6,5.62,9.936,7.923,7.103,3.793,3.791,3.791,3.791,3.791,3.79,3.916,3	1440	0012
13	7,91,3.908,3.907,3.906,3.905,4.075,4.043,4.031,4.025,4.022,4.02,4.3	1441	0013
14	86,4.236,4.192,4.172,4.161,4.183,4.933,4.581,4.457,4.401,4.369,4.34	1442	0014
15	96,6.03,5.21,4.92,4.791,4.715,4.759,7.93,6.276,5.692,5.433,5.279,	1443	0015
16	DAT AAB/5.173,3.79,3.79,3.789,3.789,3.789,4.02,4.014,4.011,4.01,4.0	1444	0016
17	108,4.346,4.29,4.265,4.25,4.24,5.173,4.912,4.796,4.726,4.679,3.47,3	1445	0017
18	2,511,3.504,3.507,3.624,3.622,3.792,3.789,3.744,4.106,4.297,4.346,3	1446	0018
19	3,658,3.842,3.961,4.015,3.537,3.637,3.708,3.749,3.479,3.535,3.579,3	1447	0019
20	4,608,3.461,3.495,3.523,3.542,3.461,3.478,3.494,3.511,2.586,2.981,3	1448	0020
21	5,396,3.693,3.806,3.785,3.434,3.187,3.036,2.861,3.089,3.309,3.498,3	1449	0021
22	6,628,3.693,3.64,3.501,3.379,3.2,3.342,3.474,3.584,3.673,3.737,3.8	1450	0022
23	73,744,3.681,3.519,3.633,3.738,3.826,3.903,3.957,4.03,4.041,4.015,3	1451	0023
24	8,744,3.838,3.921,3.994,4.06,4.11,4.223,4.248,4.243,3.877,3.951,4.	1452	0024
25	9018,4.079,4.133,4.17,4.292,4.333,4.343,3.926,3.98,4.033,4.087,	1453	0025
26	DAT AAC/4.125,4.152,4.263,4.317,4.346,1.334,1.264,1.217,1.2,1.884,1	1454	0026
27	1,742,1.667,1.587,2.337,2.151,2.025,1.947,2.871,2.506,2.337,2.232,3	1455	0027
28	2,311,2.792,2.584,2.461,3.668,3.035,2.789,2.652,3.846,3.23,2.968,2.	1456	0028
29	3821,3.84,3.37,3.113,2.963,3.767,3.475,3.25,3.104,1.543,1.515,1.491	1457	0029
30	4,1,468,1.447,1.428,1.414,2.003,1.96,1.922,1.89,1.862,1.835,1.812,2	1458	0030
31	5,51,2.426,2.357,2.301,2.253,2.21,2.173,3.151,2.967,2.836,2.734,2.6	1459	0031
32	652,2.585,2.529,4.07,3.651,3.383,3.198,3.063,2.956,2.871,5.417,4.48	1460	0032
33	76,3.997,3.69,3.477,3.318,3.196,5.793,5.081,4.509,4.119,3.844,3.64,	1461	0033
34	83,484,4.855,4.893,4.637,4.339,4.084,3.879,3.714,4.14,4.37,4.391,4.	1462	0034
35	9286,4.134,3.98,3.841,3.735,3.956,4.069,4.083,4.035,3.954,3.863,	1463	0035
36	DAT AAD/3.505,3.692,3.819,3.881,3.891,3.868,3.824,3.378,3.516,3.654	1464	0036
37	1,3.712,3.77,3.769,3.767,5.417,4.486,3.997,3.69,5.82,4.795,4.229,3.	1465	0037
38	2862,5.887,5.022,4.43,4.033,5.639,5.107,4.573,4.163,5.23,5.054,4.64	1466	0038
39	33,4.251,4.855,4.893,4.637,4.339,2.47,2.73,3.077,3.551,4.192,4.958,	1467	0039
40	45,659,5.946,5.793,2.482,2.678,2.916,3.205,3.549,3.916,4.304,4.637,	1468	0040
41	54,855,2.503,2.656,2.833,3.033,3.255,3.496,3.725,3.949,4.14,2.535,2	1469	0041
42	6,659,2.795,2.942,3.1,3.264,3.43,3.59,3.735,2.582,2.683,2.792,2.907	1470	0042
43	7,3.026,3.146,3.27,3.391,3.505,2.641,2.731,2.822,2.913,3.003,3.097,	1471	0043
44	83,191,3.284,3.378,25.22,30.03,35.61,40.25,44.37,48.45,24.47,30.12,	1472	0044
45	936,7.42,16,47.08,51.94,22.95,29.45,36.95,43.16,48.74,54.28,20.18,	1473	0045
46	DAT AAE/27.65,36.2,43.19,49.41,55.58,16.12,24.74,34.5,42.3,49.15,55	1474	0046
47	1,9,10.33,20.63,31.93,40.59,48.07,55.4,1.494,15.13,28.59,38.11,46,1	1475	0047
48	25,53.98,-19.29,10.62,24.91,35.43,5,51.78,0.8,28,17.76,30.56,39.5	1476	0048
49	33,48.33,0,-55.4,28,17.23,31.9,42.56,0,0,0,-9.11,3.79,17.32,36,1	1477	0049
50	41,93,-5.256,1.59,0,-11,16.88,15.48,11,4.18,-13.56,0,0,23.81,18	1478	0050
51	5,76,13.42,-1.68,-17.82,0,31.05,25.95,27.7,16.41,3.2,-10.31,38.05,	1479	0051
52	632,9,33.48,2,32.03,20.87,9.05,44.95,39.75,39.57,39.66,34.95,24.46,10	1480	0052
53	727,1148,1238,1311,1362,1407,1445,962,3,1099,1202,1285,13	1481	0053
54	844,1394,1438,869,1026,1142,1235,1301,1357,1405,727.5,906	1482	0054
55	9,9,1040,1145,1219,1282,1335,533,3,742,1,895,5,1016,1099,1	1483	0055
56	DAT AAF/1170,1230,333.8,512.8,696.3,838.4,934.9,1015,1083,991,2,	1484	0056
57	1290.2,434.2,603.4,720.3,814.5,893.8,0,0,0,153.4,302.5,433,2,558.1,	1485	0057

***** SPHTDA *****

58	2653.3,343.4,72.4,-162.20.151.5,276.4,378.8,929.7,692.1,477.8,288	1486 0058
59	3.3,134.90.2,-120.1489.1269.1082.907.6,759.5,607.4,444.7,2037.	1487 0059
60	4,1817.1634.1469.1331.1195.1053.2584.2363.2180.2016.1881.	1488 0060
61	5,1751.1620.3135.2911.2726.2563.2428.2299.2173.1566.1557	1489 0061
62	6,1.78,1.77,2.059,2.037,2.377,2.333,2.926,2.724,1.091,3.21,2.46,5.2	1490 0062
63	786,2,456,3,245,2,428,3,431,2,434,3,025,2,461,2,813,2,463,2,7,2,465	1491 0063
64	8,2.638,2,47,2,612,35.34,16.83,13.18,11.55,10.58,9.91,9.429,54.51.	1492 0064
65	926.45,20.19,17.34,15.62,14.45,13.58,64.37,38.42,29.61,25.31,22.66/	1493 0065
66	DATAAG/20.81,19.43,56.31,49.66,39.93,34.49,30.94,28.41,26.46,9.931	1494 0066
67	1,7.923,7.103,15.62,10.89,9.417,22.66,15.12,12.73,30.94,20.32,16.85	1495 0067
68	2,7.93,6.276,5.692,5.433,5.279,5.173,10.9,7.934,6.883,6.417,6.14,5.	1496 0068
69	395,15.15,10.31,8.585,7.82,7.364,7.052,17.43,13.27,10.71,9.57,8.89,	1497 0069
70	48.424,5.173,4.912,4.796,4.726,4.679,7.052,6.278,5.935,5.731,5.591,	1498 0070
71	59.796,8.258,7.575,7.167,6.891,0.3.802,3.799,3.797,3.797,3.796,3.8	1499 0071
72	686,3.875,3.87,3.867,3.865,4.01,3.979,3.965,3.956,3.951,4.216,4.14,	1500 0072
73	74.106,4.085,4.072,4.576,4.408,4.333,4.289,4.259,5.191,4.855,4.707,	1501 0073
74	84.618,4.558,6.202,5.582,5.308,5.144,5.032,7.785,6.713,6.238,5.954,	1502 0074
75	95.761,10.15,8.396,7.619,7.155,6.839,13.53,10.8,9.588,8.865,8.372/	1503 0075
76	DATAAH/18.14,14.09,12.29,11.21,10.47,24.18,18.44,15.86,14.32,13.26	1504 0076
77	1,31.64,23.93,20.41,18.28,16.82,40.2,30.53,25.96,23.16,21.23,48.9,3	1505 0077
78	27.93,32.38,28.89,26.45,55.98,45.39,39.27,35.23,32.33,59.18,51.55,4	1506 0078
79	35.76,41.58,38.42,57.08,54.85,50.68,47.04,44.04/	1507 0079
80	DATAAI/2.819,2.811,2.809,2.809,2.808,2.808,2.808,3.032,2.963,2.95,	1508 0080
81	12.944,2.94,2.938,2.936,3.624,3.255,3.186,3.155,3.136,3.124,3.114,5	1509 0081
82	2.313,3.972,3.717,3.604,3.537,3.491,3.457,9.333,5.61,4.898,4.582,4.	1510 0082
83	3394,4.265,4.17,17.15,8.823,7.203,6.481,6.049,5.754,5.536,29.47,14.	1511 0083
84	422,11.11,9.705,8.862,8.284,7.855,2.808,2.807,2.807,2.938,2.931,2.9	1512 0084
85	528,3.124,3.086,3.07,3.492,3.353,3.296,4.268,3.878,3.719,5.762,4.86	1513 0085
86	65,4,498,8.3,6,533,5.809,2.807,2.807,2.806,2.806,2.806,2.806,2.931,	1514 0086
87	72.925,2.923,2.922,2.921,2.92,3.085,3.055,3.044,3.039,3.036,3.034,3	1515 0087
88	8.353,3.239,3.199,3.181,3.17,3.189,3.879,3.559,3.446,3.395,3.365,3.	1516 0088
89	9345,4.867,4.129,3.867,3.751,3.682,3.72,6.541,5.077,4.557,4.327/	1517 0089
90	DATAAJ/4.189,4.095,2.806,2.806,2.805,2.804,2.803,3.034,3.029,3.026	1518 0090
91	1,3.024,3.022,3.345,3.294,3.271,3.257,3.247,4.095,3.861,3.757,3.694	1519 0091
92	2,3.652,2.485,2.509,2.519,2.52,2.639,2.637,2.807,2.803,2.884,2.996,	1520 0092
93	33.078,3.134,2.728,2.802,2.86,2.9,2.578,2.626,2.663,2.689,2.506,2.5	1521 0093
94	438,2.562,2.578,2.482,2.504,2.52,2.53,2.48,2.489,2.5,2.509,1.602,1.	1522 0094
95	5634,1.665,1.682,1.694,1.706,1.758,1.8,1.838,1.876,1.893,1.911,1.92	1523 0095
96	69,1.947,1.961,2.005,2.046,2.084,2.215,2.216,2.217,2.219,2.229,2.23	1524 0096
97	78,2.271,2.304,2.338,2.533,2.543,2.552,2.56,2.571,2.58,2.617,2.653,	1525 0097
98	82.684,2.759,2.776,2.791,2.806,2.82,2.833,2.882,2.92,2.951,2.892,2.	1526 0098
99	991,2.926,2.943,2.958,2.972,3.023,3.062,3.093,2.94,2.956,2.973/	1527 0099
100	DATAAK/2.989,3.003,3.015,3.063,3.101,3.134,1.113,1.083,1.06,1.05,1	1528 0100
101	1.353,1.335,1.325,1.312,1.496,1.498,1.494,1.491,1.572,1.591,1.601,1	1529 0101
102	2.603,1.595,1.63,1.655,1.672,1.617,1.662,1.696,1.724,1.651,1.699,1.	1530 0102
103	3738,1.771,1.693,1.741,1.782,1.819,1.757,1.807,1.85,1.888,1.173,1.1	1531 0103
104	471,1.168,1.165,1.16,1.156,1.149,1.345,1.341,1.34,1.338,1.334,1.332	1532 0104
105	5,1.329,1.456,1.455,1.454,1.453,1.452,1.45,1.449,1.517,1.519,1.521,	1533 0105
106	61.523,1.525,1.526,1.527,1.551,1.552,1.536,1.561,1.564,1.569,1.572,	1534 0106
107	71.583,1.571,1.571,1.575,1.581,1.587,1.531,1.61,1.591,1.584,1.584,1	1535 0107
108	8.589,1.594,1.6,1.61,1.61,1.608,1.608,1.61,1.615,1.621,1.616,1.623,	1536 0108
109	91.627,1.632,1.635,1.641,1.646,1.632,1.63,1.646,1.653,1.659,1.665/	1537 0109
110	DATAAL/1.671,1.665,1.673,1.68,1.687,1.633,1.7,1.707,1.715,1.723,1.	1538 0110
111	1731,1.738,1.744,1.751,1.757,1.583,1.571,1.571,1.575,1.597,1.578,1.	1539 0111
112	2574,1.579,1.608,1.587,1.581,1.582,1.61,1.596,1.589,1.589,1.61,1.60	1540 0112
113	34,1.598,1.598,1.61,1.61,1.608,1.608,1.485,1.511,1.534,1.557,1.577,	1541 0113
114	41.604,1.621,1.621,1.61,1.495,1.516,1.534,1.552,1.569,1.583,1.596,1	1542 0114
115	5.606,1.61,1.516,1.532,1.548,1.563,1.577,1.591,1.602,1.61,1.616,1.5	1543 0115

***** SPHTDA *****

116	649.1.562.1.574.1.585.1.597.1.608.1.619.1.627.1.632.1.595.1.606.1.6	1544 01
117	715.1.625.1.634.1.642.1.652.1.66.1.665.1.656.1.664.1.671.1.679.1.68	1545 01
118	86.1.693.1.701.1.708.1.715.1.731.1.732.1.733.1.733.1.731.1.24	1546 01
119	9.1.239.1.238.1.237.1.236.1.235.1.342.1.338.1.335.1.332.1.33.1.328/	1547 01
120	DATAH/1.417.1.412.1.409.1.406.1.404.1.402.1.464.1.462.1.46.1.459.	1548 01
121	11.457.1.456.1.501.1.497.1.495.1.495.1.495.1.565.1.535.1.523.	1549 01
122	21.52.1.52.1.522.1.754.1.78.1.564.1.546.1.541.1.541.1.943.2.3.1.71.	1550 01
123	31.585.1.564.1.557.0.2.4.1.808.1.653.1.594.1.573.0.0.1.62.1.683.	1551 01
124	41.624.1.591.1.583.2.3.3.1.0.1.632.1.608.1.33.1.976.2.608.1.56.0.	1552 01
125	50.1.522.1.66.1.736.1.617.1.495.0.1.527.1.589.1.679.1.663.1.609.1	1553 01
126	6.56.1.522.1.563.1.608.1.648.1.627.1.601.1.52.1.551.1.579.1.613.1.6	1554 01
127	722.1.609.1.151.1.152.1.152.1.152.1.153.1.153.1.153.1.243.1.242.1.2	1555 01
128	841.1.241.1.24.1.24.1.24.1.327.1.326.1.325.1.324.1.323.1.322.1.321.	1556 01
129	91.393.1.392.1.391.1.39.1.389.1.388.1.387.1.44.1.439.1.438.1.438/	1557 01
130	DATAAN/1.437.1.436.1.436.1.478.1.474.1.473.1.472.1.471.1.471.1.47.	1558 01
131	11.508.1.503.1.506.1.504.1.501.1.5.1.498.0.0.1.547.1.546.1.548.1.	1559 01
132	2539.1.533.1.646.1.78.1.938.1.602.1.615.1.628.1.641.1.544.1.607.1.6	1560 01
133	39.1.805.1.978.2.346.2.652.1.517.1.544.1.578.1.623.1.687.1.787.1.96	1561 01
134	42.1.504.1.518.1.534.1.551.1.57.1.592.1.618.1.498.1.508.1.517.1.526	1562 01
135	5.1.536.1.546.1.556.1.499.1.506.1.512.1.519.1.526.1.532.1.539.1.129	1563 01
136	6.1.13.1.22.1.22.1.31.1.307.1.38.1.378.1.432.1.43.1.477.1.469.1.477	1564 01
137	7.1.7.1.475.1.671.1.476.1.619.1.477.1.544.1.478.1.517.1.478.1.504.1	1565 01
138	8.481.1.498.1.485.1.499.29.47.14.22.11.11.9.705.8.862.8.284.7.855.4	1566 01
139	93.57.21.93.16.86.14.51.13.08.12.09.11.36.50.02.30.81.24.1.20.75/	1567 01
140	DATAAO/18.65.17.17.16.06.44.26.38.61.31.58.27.55.24.89.22.97.21.48	1568 01
141	1.8.3.6.533.5.809.13.08.9.068.7.803.18.65.12.54.10.56.24.89.16.66.1	1569 01
142	23.88.6.61.1.5.077.4.597.4.327.4.189.4.095.10.09.6.517.5.6.5.191.4.9	1570 01
143	347.4.78.12.6.8.52.7.052.6.395.6.001.5.732.16.8.10.95.8.824.7.865.7	1571 01
144	4.289.6.894.4.095.3.861.3.757.3.694.3.652.5.732.5.06.4.761.4.584.4.	1572 01
145	546.8.056.6.756.6.173.5.816.5.449.0.2.817.2.813.2.812.2.811.2.811.	1573 01
146	62.899.2.888.2.883.2.881.2.879.3.018.2.989.2.976.2.968.2.963.3.214.	1574 01
147	73.142.3.111.3.092.3.079.3.55.3.394.3.325.3.284.3.255.4.119.3.809.3	1575 01
148	8.672.3.59.3.534.5.044.4.476.4.225.4.074.3.972.6.478.5.504.5.071.4.	1576 01
149	9813.4.637.8.587.7.014.6.314.5.895.5.609.11.54.9.137.8.061.7.417/	1577 01
150	DATAAP/6.976.15.49.11.99.10.42.9.471.8.822.20.48.15.66.13.46.12.13	1578 01
151	1.11.22.26.41.20.15.17.23.15.45.14.22.32.93.25.35.21.67.19.4.17.81.	1579 01
152	239.24.30.92.26.61.23.87.21.93.44.14.36.29.31.69.28.62.26.38.46.18.	1580 01
153	340.53.36.27.33.17.30.8.44.47.42.64.39.57.36.91.34.71/	1581 01
154	END	1582 01

***** PROCEDURE DEFINITION PRØCESSØR - SPUMP

```
1 SPUMP* PROC
2 C
3 REAL NSG,NSS,NUZ,NU
4 C
5 COMMON /SPUMP/ DI,EFFQ, H, DNS, XNS, NSG, NSS, NUZ, NU(2), U
6 C
7 END
```

***** SUBROUTINE STOCON

```

1      C      *****
2      C      * ROUTINE NAME = ROUTINE TO PACK THE FIRST *
3      C      * WORD OF THE CONFIGURATION *
4      C      * TABLE. *
5      C      * ROUTINE LANG = FORTRAN V UNIVAC 1108 EXEC 2*
6      C      * PROGRAMMER   = R. BOLLINGER 1943 102 26933 *
7      C      * DATE CODED   = 3/11/70 *
8      C      *****
9      C
10     C      SUBROUTINE STOCON(IDX)
11     C
12     C      ***** EXPLANATION OF THE CALLING SEQUENCE
13     C      *
14     C      ***** IDX = INDEX OF THE CONFIGURATION TABLE.
15     C
16     C      INCLUDE CCNFIG
17     C
18     C      DO 10 I1 = 1,6
19     C      IF(ICNFIG(I1).EQ.0) ICNFIG(I1) = 0
20     C      CONFIG(IDX,I1) = GPRYTE(ICNFIG(I1),6,CONFIG(IDX,I1),I1)
21     C  10 CONTINUE
22     C
23     C      RETURN
24     C      END

```

***** SUBROUTINE STODTA

```

1  C
2  SUBROUTINE STODTA
3  C
4  INCLUDE CAPU
5  INCLUDE CCONFIG
6  INCLUDE CCNTRL
7  INCLUDE CECLSS
8  INCLUDE CFLRAT
9  INCLUDE CHEX
10 INCLUDE CHSORC
11 INCLUDE CIOUNT
12 INCLUDE CMATRL
13 INCLUDE CNAMES
14 INCLUDE CONST
15 INCLUDE CPAGE
16 INCLUDE CPUMP
17 INCLUDE CTANK
18 INCLUDE CTURBN
19 INCLUDE TABLOK
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C

```

***** PDP CAPU *****

```

DATA ((LAPU1(I,J),I=1,4),J=1,8) / 'PROPELLANT TEMP. (DEG-R)',
1 'OXYGEN LOADED (LBS)' , 'HYDROGEN LOADED (LBS)',
2 'TOTAL LOADED (LBS)' , 'PROP. USED BY APU (LBS)',
3 'OXY. MAX. FLOW (LB/SEC)' , 'HYD. MAX. FLOW (LB/SEC)',
4 'WDOT OX-TURB-GG (LB/SEC)' /

```

***** PDP CCNFIG *****

```

DATA ((LCNFI(I,J),I=1,4),J=1,6) / 'CONSUMER WEIGHT - LBS' ,
1 'OXIDIZER SYSTEM WT. -LBS' , 'OXID INSULATION WT - LBS' ,
2 'FUEL SYSTEM WT. - LBS' , 'FUEL INSULATION WT - LBS' ,
3 'TOTAL SYSTEM WT. - LBS' /

```

***** PDP CCNTRL *****

```

DATA ((INBLK(1,I,J),I=1,5),J=1,2) / SUB CRIT. SUPER CRIT.
1,1,1,1,0. 1,1,0,1,0 /
DATA ((INBLK(2,I,J),I=1,5),J=1,2) / 1,1,1,1,1. 1,1,0,1,0 /
DATA ((INBLK(3,I,J),I=1,5),J=1,2) / 0,0,0,0,0. 0,1,0,0,0 /
DATA ((INBLK(4,I,J),I=1,5),J=1,2) / 0,1,0,1,0. 0,1,1,0,1 /
DATA ((INBLK(5,I,J),I=1,5),J=1,2) / 0,0,0,0,0. 0,0,1,0,0 /

```



```

***** STODTA *****
58 DATA (KSUBC(1,I),I=1,NBRSR) / 6, 4, 10, 9, 8, 1, 10, 11, 2/
59 DATA (KSUBC(2,I),I=1,NBRSR) / 6, 3, 4, 10, 11, 2, 0, 0, 0/
60 DATA (KSUBC(3,I),I=1,NBRSR) / 7, 0, 0, 0, 0, 0, 0, 0, 0/
61 DATA (KSUBC(4,I),I=1,NBRSR) / 5, 4, 0, 0, 0, 0, 0, 0, 0/
62 DATA (KSUBC(5,I),I=1,NBRSR) / 6, 4, 10, 9, 8, 1, 10, 11, 2/
63 DATA JAPUS(1,1),JAPUS(1,2) / 4, 3/
64 DATA JAPUS(2,1),JAPUS(2,2) / 3, 4/
65 DATA NAMSYS / 'ACPI','APUI','EC','FUE','OMS' /
66 C
67 C ***** PDP CFLRAT *****
68 C
69 DATA ((LFRT(I,J),I=1,3),J=1,7) /
70 1 'WDOT OX-TURB.-G.G.', 'WDOT HY-TURB.-G.G.', 'WDOT BOTH TURB.-G.G.',
71 2 'WDOT OXY HEX.-G.G.', 'WDOT HYD HEX.-G.G.', 'WDOT BOTH HEX.-G.G.',
72 3 'TOTAL FLOWRATE **' /
73 C
74 C ***** PDP CHEX *****
75 C
76 DATA ((UNAH(I,J),I=1,2),J=1,4) /
77 1 'BOILING', 'SUP-CRITICAL', 'PARALLEL-FLOW', 'COUNTER-FLOW' /
78 DATA ((LHX1(I,J),I=1,4),J=1,10) /
79 1 'THERM CONDUCTANCE RATIO', 'HOT FLUID FLOW RATE',
80 2 'COLD FLUID DELTA - P', 'CAPACITY RATIO',
81 3 'NUMBER OF TRANSFER UNITS', 'COMPUTED VALUE OF UA',
82 4 'COMPUTED VALUE OF W/UA', 'WEIGHT OF SUBUNIT',
83 5 'WEIGHT OF HEAT EXCHANGER', 'HEX SUBUNIT TYPE **' /
84 DATA ((LHX2(I,J),I=1,4),J=1,14) /
85 1 'COLD FLUID INLET TEMP', 'COLD FLUID OUTLET TEMP',
86 2 'COLD FLUID SPECIFIC HEAT', 'COLD FLUID FLOW RATE',
87 3 'HOT FLUID INLET TEMP', 'HOT FLUID OUTLET TEMP',
88 4 'HOT FLUID SPECIFIC HEAT', 'HOT FLUID FLOW RATE',
89 5 'COLD SIDE EFFECTIVENESS', 'HOT SIDE EFFECTIVENESS',
90 6 'TOTAL EFFECTIVENESS', 'HEAT EXCHANGER UA/A-WALL',
91 7 'HEAT EXCHANGER DIAMETER', 'HEAT EXCHANGER LENGTH' /
92 DATA LHX3 / 'HEAT EXCHANGER CHARACTERISTICS' /
93 C
94 C ***** PDP CHSorc *****
95 C
96 DATA ((LHS1(I,J),I=1,5),J=1,6) /
97 1 'GAS GENERATOR CHARACTERISTICS', 'GAS GEN. FLOW RATE - (LB/SEC)',
98 2 'GAS GEN. PROPELLANT WGT. - (LBS)', 'GAS GENERATOR WEIGHT - (LBS)',
99 3 'CUMULATIVE GAS GEN. PROP. WGT.', 'WEIGHT OF HEX-GAS GEN. ASSY.' /
100 C
101 DATA ((LHS2(I,J),I=1,5),J=1,14) /
102 1 'SPEC. HEAT AVAILABLE - (BTU/LB-R)', 'TOTAL HEAT REQUIRED - (BTU)',
103 2 'HOT FLUID REQUIRED - (LBS)', 'CUMULATIVE HEAT REQD. - (BTU)',
104 3 'CUMULATIVE HOT FLUID - (LBS)', 'WASTE HEAT UTILIZATION DATA',
105 4 'MAX HOT FLUID FLOW-RATE (LBS/HR)', 'CYCLE MAX REQD ENERGY - O2 HEX',
106 5 'CYCLE MAX REQD ENERGY - N2 HEX', 'CYCLE MAX REQD ENERGY - O2 TANK',
107 6 'CYCLE MAX REQD ENERGY - N2 TANK', 'TOTAL MAX ENERGY - HEX + TANKS',
108 7 'TOTAL ENERGY FOR MISSION SPAN', 'TOTAL ENERGY REQMT - KW/HRS' /
109 C
110 C ***** PDP CIOUNT *****
111 C
112 DATA IOUNIT/14,21,22,23,19,29,15,16,17,18,25,26,27,28/
113 DATA IIN,IOT / 5,6 /
114 C
115 C ***** PDP CMATRL *****

```

***** STODTA *****

```

116 C
117 DATA (RHOL(I),I=1,5)/501.120,176.256,169.344,511.488,276.480/
118 DATA (RHOI(I),I=1,9)/2.14,2.45,2.80,0.836,0.59,0.65,5.0,2.2,1.0/
119 DATA (RHOIS(I),I=1,9)/0.0428,0.0490,0.0280,0.027867,0.01475,
120 1 0.021667,5.0,2.20,1.0/
121 DATA (MINTHK(I),I=1,15)/0.020,0.025,0.028,0.020,0.016,0.035,0.058,
122 1 0.065,0.042,0.020,0.049,0.083,0.095,0.049,0.035/
123 C
124 C ***** PDP CNAMES *****
125 C
126 DATA (FNAME(I),I=1,18)/'GAS 1','ENGINE 1','LINE 1','CONTRL 1','FITING 1'
127 1 'TAP 1','TEE 1','ELBOW 1','VALVE 1','REG 1'
128 2 'ACCUM 1','TANK 1','PUMP 1','HEX 1','TRBINE 1'
129 3 'F-CELL 1','EC/LSS 1','END 1'
130 C
131 DATA ((LO(I,J),I=1,9),J=1,15)/
132 1 ***** SYSTEM CONFIGURATION *****
133 2 ***** ENGINE DATA *****
134 3 ***** ACCUMULATOR DATA *****
135 4 ***** HEAT EXCHANGER DATA *****
136 5 ***** HIGH PRES PUMP DATA *****
137 6 ***** DUTY CYCLE DATA *****
138 7 ***** HEAT SOURCE DATA *****
139 8 ***** TURBINE DATA *****
140 9 ***** MOTOR DATA *****
141 T ***** TANK CONFIGURATION DATA *****
142 1 ***** LOW PRES PUMP DATA *****
143 2 ***** TANK DATA *****
144 3 ***** AUXILIARY POWER UNIT *****
145 4 ***** FUEL CELL DATA *****
146 5 ***** EC / LSS DATA *****
147 C
148 DATA ((L1(I,J),I=1,2),J=1,2)/' COMP COMP FUNC. NUMB. NU
149 1 MB. MATRL. FLOW FRICTION LINE LENGTH LINE INSULATION
150 2 INSULATION NO. LAYERS 1, NAME CODE TYPE OPER. STBY
151 3 TYPE COEFFICIENT OR L-OVER-D DIAMETER TYPE T
152 4 THICKNESS INSULATION 1/
153 C
154 DATA ((L2(I,J),I=1,3),J=1,7)/'NUMBER OF ENGINES 1,
155 1 'GAS INLET TEMP. 1,'GAS INLET PRES. 1,
156 2 'ENGINE THRUST 1,'CHAMBER PRES. 1,
157 3 'EXPANSION RATIO 1,'MIXTURE RATIO 1/
158 C
159 DATA ((L3(I,J),I=1,4),J=1,3) / 'OPERATING TEMP. (DEG R) 1,
160 1 'TANK VOLUME (CU. FT.) 1, 'NOMINAL OPER. DELTA PRES'1/
161 C
162 DATA ((L4(I,J),I=1,4),J=1,11)/ 'HEX HOT INLET TEMP. 1,
163 1 'HEX HOT OUTLET TEMP. 1,'HEX COLD INLET TEMP. 1,
164 2 'HEX COLD OUTLET TEMP. 1,'HEX HOT INLET PRES. 1,
165 3 'HEX HOT OUTLET PRES. 1,'HEX COLD INLET PRES. 1,
166 4 'HEX COLD OUTLET PRES. 1,'HEX HOT SIDE DELTA-P 1,
167 5 'HEX COLD SIDE DELTA-P 1,'HEX GAS GEN. O/F RATIO 1/
168 C
169 DATA ((L5(I,J),I=1,4),J=1,5)/ 'TYPE 1,
170 1 'EFFICIENCY 1,'NET + SUCTION HEAD 1,
171 2 'SHAFT SPEED 1,'ESTIMATED DELTA PRES. 1/
172 C
173 DATA ((L6(I,J),I=1,4),J=1,4)/ 'PUMP EFFICIENCY 1,

```

B-271

LMSC-A991396

B-272

```

***** STODTA *****
232 2 'DAYS SUPPLY RESERVE GAS 1,'O2 CONSUMED PER MAN-DAY 1,
233 3 'VEHICLE GAS LEAKAGE RATE1,'DELIVERED O2 NOM. TEMP. 1,
234 4 'DELIVERED N2 NOM. TEMP. 1,'O2 FILL DENSITY 1,
235 5 'N2 FILL DENSITY 1,'O2 TANK FINAL TEMP 1,
236 6 'N2 TANK FINAL TEMP. 1,'O2 TANK FINAL PRESSURE 1,
237 7 'N2 TANK FINAL PRESSURE 1,'LSS ENVIRONMENT TEMP. 1,
238 8 'CABIN OR AIRLOCK VOLUME 1,'O2-HEX INLET LINE DIAM. 1,
239 9 'N2-HEX INLET LINE DIAM. 1,'HEX HEATER ENERGY RATING1,
240 T 'TNK HEATER ENERGY RATING1,'DELIVERED O2 PRESSURE 1,
241 A 'DELIVERED N2 PRESSURE 1,'O2 TANK HEATER DIAMETER 1,
242 B 'N2 TANK HEATER DIAMETER 1,'O2 TANK HEATER LENGTH 1,
243 C 'N2 TANK HEATER LENGTH 1,'O2 TANK LOW-PRESS. LIMIT1,
244 D 'N2 TANK LOW-PRESS. LIMIT1/
245 C
246 DATA ((JFLUID(I,J),I=1,2),J=1,3)/ OXYGEN 1,1 HYDROGEN 1,
247 1 ' NITROGEN 1/
248 DATA ((KFLUID(I,J),I=1,2),J=1,2)/ OXIDIZER 1,1 FUEL 1/
249 C
250 ***** PDP CONST *****
251 C
252 DATA GRAVTY,PI,PI203 / 32.172 ,3.14159265 ,2.0943951 /
253 C
254 ***** PDP CPAGE *****
255 C
256 DATA MAXLIN,UNUM ,OPTLUN/50,'AT4307',6 /
257 DATA PTITLE/' THE INTEGRATED MATH MODEL 1/
258 C
259 ***** PDP CPUMP *****
260 C
261 DATA ((LPP1(I,J),I=1,3),J=1,6) /
262 1'TEMPERATURE 1,'PRESSURE 1,'FLOW RATE 1,
263 2'DELTA-PRESSURE 1,'NPSH AVAILABLE 1,'DENSITY OF FLUID 1/
264 DATA LPP2 / 'NUMBER OF STAGES REQD. 1 /
265 DATA ((LPP3(I,J),I=1,3),J=1,6) /
266 1'COMPUTED NPSH REQD1,'COMPUTED PUMP EFF.1,'COMPUTED PUMP VOL.1,
267 2'COMPUTED PUMP WGT.1,'COMPUTED PUMP PWR.1,'COMPUTED PUMP SPD.1/
268 DATA LPP4 / 'SELECTED PUMP OPTION 1 /
269 DATA LPP5 / 'PUMP CHARACTERISTICS 1/
270 C
271 ***** PDP CTANK *****
272 C
273 DATA ((LTZ1(I,J),I=1,3),J=1,14)/
274 1 'NUMBER OF TANKS 1,'MATERIAL TYPE 1,'INSULATION TYPE 1,
275 2 'FLUID WGT. (TOTAL)1,'FLUID VOLUME /TANK1,'DIAMETER (FT)/TANK1,
276 3 'SURFACE AREA /TANK1,'TANK VOLUME / TANK1,'TANK WGT. (LB) TOT1,
277 4 'INSUL. THICKNESS 1,'INSUL. WT (LB) TOT1,'HEAT LEAK BTU/H/FT1,
278 5 'GAS RESIDUALS WT. 1,'HGT ADDED CYL SECT1/
279 DATA ((LTZ2(I,J),I=1,2),J=1,3)
280 1 / 'SURF TENSION1, 'POSITV DISPL1, 'DYELECTROPHORI1/
281 DATA ((LTZ3(I,J),I=1,3),J=1,4) /'TYPE ACQ. DEVICE. 1,
282 1 'DEVICE WT. (LBS) 1,'TRAPPED BY DEVICE 1,'RESID. PROPELLANT1/
283 C
284 ***** PDP CTURBN *****
285 C
286 DATA LTBN1 /'TURBINE CHARACTERISTICS 1/
287 DATA ((LTBN2(I,J),I=1,5),J=1,6) /
288 1'TURBINE ROTOR MEAN DIAMETER 1,'WGT. OF PWR. TRANSMISSION ASSY1,
289 2'WGT. OF TURBINE ROTOR 1,'WGT. OF MANIFOLD AND NOZZLE 1,

```

```

***** STODTA *****
290      3*WEIGHT OF INDUCER      1*WEIGHT OF TURBINE ASSY.
291      C
292      C      ***** PDP TABLOK *****
293      C
294      DATA NTBID / 1, 2, 7, 8, 4, 3, 6, 5, 21, 9,
295      1      10, 10, 17, 13, 13, 16, 13, 21, 27, 28,
296      2      29, 30, 31, 32, 21, 32, 35, 32, 38, 35,
297      3      10, 13, 16, 13, 41, 41, 32, 32, 41, 18,
298      4      19, 20, 21, 44, 45, 46, 0, 0, 0, 0/
299      MIPE = 0.
300      NIENH = 0.0
301      C
302      RETURN
303      C
304      END

```

PROCEDURE DEFINITION PRØCESSØR - TABLØK

```
1 TABLØK* PROC
2 C
3     REAL MIPE
4 C
5     COMMON /TABLØK/ XTAB(7), NTBID(50)
6 C
7     END
```

SUBROUTINE TANK

```

1      C      * * * * *
2      C      * ROUTINE NAME - TANK PRES. AND WT. DUTY *
3      C      * CYCLE HISTORY *
4      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
5      C      * PROGRAMMER - R. VERNON, 6263, 562, 24385 *
6      C      * DATE CODED - 5/12/72 *
7      C      * * * * *
8      C
9      C      SUBROUTINE TANK
10     C
11     C      LOGICAL JP, DIAG, PAGE
12     C
13     C      INCLUDE CACCUM
14     C      INCLUDE CDCYCL
15     C      INCLUDE CENG
16     C      INCLUDE CHEX
17     C      INCLUDE CIOUNT
18     C      INCLUDE CHATRL
19     C      INCLUDE CMOTOR
20     C      INCLUDE CNAME$
21     C      INCLUDE CTANK
22     C      INCLUDE TABLOK
23     C
24     C      DIMENSION PRES(1CDL1,2,1)
25     C      DIMENSION PWTOT(2)
26     C      DIMENSION THED(2)
27     C      DIMENSION WDOTX(NHX,2)
28     C
29     C      EQUIVALENCE (WDOTCF,WDOTX)
30     C
31     C      DATA THED(1), THED(2)/170.0,40.0/
32     C
33     C      ***** INITIALIZE THE PROGRAM COUNTERS
34     C      *
35     C      * IG - GAS TYPE FLAG
36     C      * IG = 1 FOR O2
37     C      * 2 FOR H2
38     C      *
39     C      * IT = 1 (ONLY) REVISED 6-30-72
40     C      *
41     C      * IP - COUNTER FOR THOSE VARIABLE STORED FOR EACH ELEMENT
42     C      * OF THE DUTY CYCLE, TANK PRES., HE PRES
43     C      *
44     C      * IW - COUNTER FOR THOSE VARIABLE STORED FOR EACH COAST
45     C      * OF THE DUTY CYCLE, VENTED GAS WEIGHT,
46     C      *
47     C      * IF - COUNTER FOR THOSE VARIABLE STORED FOR EACH BURN
48     C      * OF THE DUTY CYCLE, HE FLOW RATE, PRESSUREANT GAS
49     C      * FLOW RATE, ENERGY FLOW.
50     C      *****
51     C      IF (DIAG(0,6HTANK )) WRITE (107,6000) NDCYCL,NOP,SATYPE,SITYPE,
52     C      1 SMTYPE,SPTYPE,SMDIAM,SVLFLD,SULGPC,SHFLUX,SITEMP,SIPRES,SVPRES,
53     C      2 SITHIK,SOPRES,SPGTEM,SHDELP,SPDELP,SHOTEM,SGGPC,SGOTEM,SGHRAT
54     C
55     C      IG = 0
56     C      IT = 1
57     C      30 IG = IG + 1

```

```

***** TANK *****
58      IBURN = 0
59      ICOAST = 0
60      IF = 0
61      IP = 0
62      ISW = 0
63      IW = 0
64      WLRT(IG,IT)=0.
65      WHESUM = 0.0
66      GO TO (80,80,70),IG
67      C
68      70 JP = DIAG(1,6HTANK )
69      RETURN
70      C
71      C      ***** CALCULATE THE EFFECTIVE TANK DENSITY RHO.
72      C
73      80 T=TSAT(SIPRES(IG,IT),IG)
74      PWTOT(IG) = WPTOT(IG)
75      WP = WPTOT(IG)
76      CALL RHOLIQ(T,IG,RHOLI)
77      VLIQ=WP/RHOLI
78      PVOL=SVOL(IG,IT)-VLIQ
79      C      CALC. RHO OF GAS
80      CALL GSDNST (IG,T,SIPRES(IG,I),RHOG)
81      WPV=RHOG*PVOL
82      WPT=WP+WPV
83      RHO=WPT/SVOL(IG,IT)
84      RHOP=RHO
85      PHE=0.
86      PPV=SIPRES(IG,IT)
87      WHE=0.
88      WTOT = WPT
89      C
90      IF(DIAG(2,6HTANK-0)) WRITE (6,6005) T,WP ,WP,RHOLI,VLIQ,
91      I SVOL(IG,IT),PVOL,RHOG,PPV,WTOT
92      C
93      C      ***** LOOK UP THE INITIAL ENERGY LEVEL
94      C
95      KTAB = 0
96      IF (IG .EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
97      CALL FINTAB (NTBID(26)+IG+KTAB)
98      XTAB(1) = RHO
99      XTAB(2) = SIPRES(IG,IT)
100     ENERGY = MIPE(2,XTAB)
101     C
102     IF(DIAG(2,6HTANK1 ))WRITE (6,6033) IG,IT,IP,IW,IF,
103     I RHOP,ENERGY,WP,WPV,WPT,SIPRES(IG,IT)
104     C
105     JP = PAGE(0)
106     C
107     WRITE (IOT,7000)
108     WRITE (IOT,7001)
109     WRITE (IOT,7002) (JFLUID(I,IG),I=1,2), T, SIPRES(IG,IT), WP, WPV,
110     I WPT, WHE, WTOT, VLIQ, PPV, PHE, PVOL, SVOL(IG,IT)
111     I 2 ,RHOP, ENERGY
112     C
113     C      ***** DO ENERGY BALANCE FOR FIRST COAST FOR ALL FOUR SYSTEMS,
114     C      ***** AND ALL COASTS FOR SYSTEM 1 (SELF PRESSURIZATION).
115     C

```



```

***** TANK *****
116 90 IP = IP + 1
117 IF(IP.LT.NDCYCL) GO TO 110
118 IF(IP.EQ.NDCYCL) GO TO 115
119 100 ISPTYP = SPTYPE(IG,IT)
120 IP = IP - 1
121 ICOAST = 0
122 IBURN = 0
123 GO TO 1010
124 C
125 110 ENERGY = ENERGY + SHRATE(IG,IT)*DCYCLE(IP+1)/WPT
126 C
127 ICOAST = ICOAST + 1
128 IF(DIAG(2,6HTANK-A)) WRITE(6,6030) ICOAST,ISW
129 C
130 IF(PAGE(35)) WRITE (IOT,7017)
131 C
132 WRITE (IOT,6030) ICOAST, ISW
133 WRITE (IOT,7003)
134 C
135 GO TO 116
136 C
137 115 ENERGY = ENERGY + SHRATE(IG,IT) * 300.0/WPT
138 IF(DIAG(2,6HTANK-F)) WRITE (6,6031)
139 C
140 IF(PAGE(17)) WRITE (IOT,7017)
141 C
142 WRITE (IOT,6031)
143 WRITE (IOT,7013)
144 C
145 116 CONTINUE
146 C
147 ***** LOOK UP RESULTING PRES.
148 C
149 CVP=CSUBV(T,PPV,IG)
150 118 CVH=0.745
151 RATIO=(WPT*CVP)/(WPT*CVP+WHE*CVH)
152 KTAB = 0
153 IF (IG .EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
154 CALL FINTAB (NTBID(27)+IG*KTAB)
155 XTAB(1)=RHOP
156 XTAB(2)=ENERGY*RATIO
157 PPV=MIPE(2,XTAB)
158 T=TSAT(PPV,IG)
159 C
160 CALC. RHO OF GAS
161 CALL GSDNST (IG,T,PPV,RHOG)
162 IF(WHE.LE.0.) GO TO 119
163 CALL ZFIND(T,PHE,17,ZHE)
164 PHE=FINDR(17)*WHE*ZHE*T/((144.*PVOL)
165 119 PRES(IP,IG,IT)=PPV+PHE
166 WPV=RHOG*PVOL
167 WR=WPT-WPV
168 C
169 IF(DIAG(2,6HTANK2 )) WRITE (6,6060) IG,IT,IP,1W,IF,
170 1 RHOP,ENERGY,CVP,CVH,RATIO,PPV,PHE,PRES(IP,IG,IT)
171 2,WP, WPV, RHOG
172 C
173 IF(DCYCLE(IP+1).LT.0.0) DCYCLE(IP+1) = 300.0

```

TANK

DATE 04

```

174 WRITE (IOT,7004) (JFLUID(I,IG),I=1,2), T, DCYCLE(IP+1), WP, WPV,
175 I WHE, PPV, PHE, PRES(IP,IG,IT),ENERGY
176 C
177 C ***** IF THE RESULTING PRES. IS GREATER THAN THE VENTING
178 C * PRES., COMPUTE THE WEIGHT OF THE VENTED MATERIAL AND
179 C ***** SET PRES TO THE VENTING PRES.
180 C
181 IN = IN + 1
182 SVWT(IN,IG,IT) = 0.0
183 IF (PRES(IP,IG,IT) - SVPRES(IG,IT)) > 130,130,120
184 120 PV=SVPRES(IG,IT)
185 PI=PRES(IP,IG,IT)
186 V=SVOL(IG,IT)
187 CALL VENT(0,,WHE,WPV,WP,T,PV,PI,V,IG,PPV,RHOP)
188 SVWT(IN,IG,IT)=WPT-WP-WPV
189 PWTOT(IG) = WP
190 WPT=WP+WPV
191 WTOT = WPT + WHE
192 PHE=PV-PPV
193 PRES(IP,IG,IT) = SVPRES(IG,IT)
194 C
195 C
196 C
197 C ***** CALCULATE A NEW TANK DENSITY
198 C
199 RHOP=WPT/SVOL(IG,IT)
200 CVP=CSUBV(T,PPV,IG)
201 128 CVH=0.745
202 RATIO=(WPT*CVP)/(WPT*CVP+WHE*CVH)
203 C
204 C
205 C ***** LOOK UP NEW ENERGY LEVEL
206 C
207 KTAB = 0
208 IF (IG.EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
209 CALL FINTAB (NTBID(28)+IG+KTAB)
210 XTAB(1)=RHOP
211 XTAB(2)=PPV
212 ENERGY=MIPE(2,XTAB)/RATIO
213 C
214 IF (DIAG(2,6HTANK3)) WRITE (6,6035) IG,IT,IP,IN,IF,
215 I SVWT(IN,IG,IT),WTOT,PPV,WP,PHE,PRES(IP,IG,IT),RHOP,RATIO,ENERGY
216 C
217 WRITE (IOT,7005)
218 WRITE (IOT,7006) SVPRES(IG,IT), SVWT(IN,IG,IT), WP, WPV, WHE,
219 I WTOT, PPV, PHE, PRES(IP,IG,IT), ENERGY
220 C
221 C ***** UPDATE THE DUTY CYCLE COUNTER AND DO THE DIFFERENT
222 C ***** SYSTEMS.
223 C
224 130 IP = IP + 1
225 IF (IP.GT.NDCYCL) GO TO 100
226 ISW = SPTYPE(IG,IT)
227 C
228 IBURN = IBURN + 1
229 IF (DIAG(2,6HTANK-B)) WRITE(6,6020) IBURN,ISW
230 C
231 WRITE (IOT,6020) IBURN, ISW

```

```

***** TANK *****
232 WRITE (IOT,7007)
233 C
234 GO TO (1000,1000,3000),ISW
235 C
236 C ***** DO ENERGY BALANCE FOR ALL BURNS
237 C
238 1000 CALL FINTAB (NTBID(29)+IG)
239 H = MIPE(1,PPV)
240 WDOTJM = WDOTJ(IBURN,IG) * PWTOT(IG)/WTTOT(IG)
241 E=ENERGY*WTTOT-H*WDOTJM*DCYCLE(IP-1)
242 C
243 IF(DIAG(2,6HTANK4A)) WRITE(6,6040) IG,IT,IP,IW,IF,
244 1 H,WDOTJ(IBURN,IG),PWTOT(IG),WTTOT(IG),WDOTJM,ENERGY,WTTOT,
245 2 DCYCLE(IP-1),E
246 C
247 WRITE (IOT,7008) (JFLUID(I,IG),I=1,2), DCYCLE(IP-1),
248 1 WDOTJ(IBURN,IG), WTTOT(IG), PWTOT(IG), ENERGY, E,
249 2 WDOTJM
250 C
251 WOUT=WDOTJM*DCYCLE(IP-1)
252 WTTOT=WTTOT-WOUT
253 WPT=WPT-WOUT
254 ENERGY=E/WTTOT
255 C
256 WTTOT(IG) = WTTOT(IG) - WDOTJ(IBURN,IG) * DCYCLE(IP-1)
257 C
258 PWTOT(IG) = WPT
259 C
260 C ***** CALCULATE NEW TANK DENSITY
261 C
262 RHOP=WPT/SVOL(IG,IT)
263 C
264 IF(DIAG(2,6HTANK4B)) WRITE(6,6050) IG,IT,IP,IW,IF,
265 1 WOUT,WTTOT,WPT,ENERGY,WTTOT(IG),PWTOT(IG),E,SVOL(IG,IT),RHOP
266 C
267 C ***** LOOK UP THE RESULTING PRESSURE
268 C
269 KTAB = 0
270 IF (IG.EQ.1) AND (RHOP.LT.40.) KTAB = 2
271 CALL FINTAB (NTBID(30)+IG+KTAB)
272 XTAB(1)=RHOP
273 RATIO=(WPT*CVP)/(WPT*CVP+WHE*CVH)
274 XTAB(2)=ENERGY*RATIO
275 PPV=MIPE(2,XTAB)
276 IF(ISW,EQ.1) PRES(IP,IG,IT)=PPV
277 C
278 IF(DIAG(2,6HTANK4C)) WRITE(6,6061) IG,IT,IP,IW,IF,
279 1 RHOP, ENERGY, RATIO, PPV, PRES(IP,IG,IT)
280 C
281 WRITE (IOT,7009)
282 WRITE (IOT,7010) WOUT, WTTOT, WPT, WTTOT(IG), RHOP, PPV, ENERGY
283 C
284 GO TO (2000,2000,3000),ISW
285 C
286 C ***** DO THE FINAL CALCULATIONS FOR SYSTEM 1 AND PART SYSTEMS 2+3
287 C
288 C ***** SUM UP THE VENTED GAS WEIGHT
289 C

```

```

***** TANK *****
290 1010 WVSUM = 0.0
291 DO 1020 I1 = 1,IW
292 WVSUM = WVSUM + SVWT(I1,IG,IT)
293 1020 CONTINUE
294 C
295 SHVTOT(IG,IT) = WVSUM
296 C
297 C ***** CALCULATE THE PROPELLANT PARENT GAS RESIDUAL.
298 C
299 TFINAL = TSAT(PPV,IG)
300 C CALC. RHO OF GAS
301 CALL GSZDNS(IG,TFINAL,PPV,RHOIG,ZIG)
302 WGR(IG,1) = RHOIG*PVOL + WHE
303 WLRT(IG,1) = WP
304 C
305 IF(DIAG(2,6HTANK5)) WRITE(6,6070) IG,IT,IP,IW,IF,WVSUM,TFINAL,Z,
306 I SHVTOT(IG,IT),WGR(IG,IT),WLRT(IG,IT)
307 C
308 WRITE (107,7014) TFINAL, SHVTOT(IG,IT), WGR(IG,IT), WLRT(IG,IT)
309 C
310 GO TO (30,2010,3010),ISPTYP
311 C
312 C ***** CALCULATE PRESSURANT NEEDED FOR THIS BURN
313 C
314 2000 T=TSAT(PPV,IG)
315 C CALC. RHO OF GAS
316 CALL GSZDNS (IG,T,PPV,RHOG,Z)
317 CALL RHOLI(T,IG,RHOLI)
318 PVOL=SVOL(IG,IT)*(RHOLI-RHOP)/(RHOLI-RHOG)
319 VLIQ=SVOL(IG,IT)-PVOL
320 WP=VLIQ*RHOLI
321 WPV=RHOG*PVOL
322 IF(ISW,EQ,1) GO TO 90
323 THE = THED(IG)
324 IF=IF+1
325 IF(PHE,LE,0.) GO TO 2004
326 CALL ZFIND(T,PHE,17,Z1)
327 RHE=WHE/PVOL
328 PHE=RHE*Z1*FINDR(17)*T/144.
329 2004 PTOT=PPV+PHE
330 IF(PTOT,LT,SOPRES(IG,IT)) GO TO 2005
331 WDOTHE(IF,IG,IT)=0.
332 WHEADD = 0.0
333 PRES(IP,IG,IT)=PTOT
334 GO TO 2006
335 2005 PHE=SOPRES(IG,IT)-ppv
336 C CALC. RHO OF GAS
337 CALL GSZDNS (17,THE,PHE,RHOGT,Z)
338 WH = RHOGT*PVOL
339 WDOTHE(IF,IG,IT)=(WH-WHE)/(DCYCLE(IP-1))
340 PRES(IP,IG,IT)=SOPRES(IG,IT)
341 WHEADD = WH - WHE
342 WHE=WH
343 2006 PRESHE(IP,IG,IT)=PHE
344 WHESUM = WHESUM + WHEADD
345 C
346 IF(DIAG(2,6HTANK6)) WRITE (6,6080) IG,IT,IP,IW,IF,
347 I T,Z,RHOG,RHOLI,PVOL,VLIQ,WP,WPV,

```

```

***** TANK *****
348      2      SOPRES(IG,IT),PPV,THE,Z),WM,WHE,
349      3      DCYCLE(IP-1),WDOTHE(IF,IG,IT),
350      4      PRES(IP,IG,IT),PRESHE(IP,IG,IT)
351      C
352      WRITE (IOT,7011)
353      WRITE (IOT,7012) T, THE, PVOL, VLIQ, WP, WPV, PRESHE(IP,IG,IT),
354      1      PTOT, SOPRES(IG,IT), WDOTHE(IF,IG,IT), WHEADD,
355      2      PRES(IP,IG,IT), WHESUM
356      C
357      GO TO 90
358      C
359      C      ***** DO FINAL CALCULATIONS FOR SYSTEM 2.
360      C
361      C      ***** CALCULATE PRESSURIZATION SYSTEM WEIGHT.
362      C
363      2010 SUMWDH = 0.0
364      DO 2020 I1 = 1,IF
365      SUMWDH = SUMWDH + WDOTHE(I1,IG,IT)*DCYCLE(2*I1-1)
366      2020 CONTINUE
367      C
368      WHETOT(IG,IT) = WHESUM
369      C
370      WPGTOT(IG,IT) = 1.5*WHETOT(IG,IT) + 40.0
371      IF(DIAG(2,6HTANK8 )) WRITE(6,6100) IG,IT,IP,IW,IF,SUMWDH,
372      1      WHETOT(IG,IT),
373      2      WPGTOT(IG,IT), WPTOT(IG), PWTOT(IG)
374      C
375      WRITE (IOT,7015)
376      WRITE (IOT,7016) WHETOT(IG,IT), WPGTOT(IG,IT)
377      C
378      GO TO 30
379      C
380      C      ***** SYSTEM 3 *****
381      C
382      C
383      C      ***** CALCULATE PRESSURANT GAS FLOW RATE.
384      C
385      3000 T = TSAT(SOPRES(IG,IT),IG)
386      IF = IF + 1
387      C      CALC. RHO OF GAS
388      CALL GSZDNS (IG,T,SOPRES(IG,I),RHOG,Z)
389      CALL RHOLIQ(T,IG,RHOLI)
390      PVOL=SVOL(IG,IT)*(RHOLI-RHOP)/(RHOLI-RHOG)
391      PRES(IP,IG,IT)=SOPRES(IG,IT)
392      PPV=PRES(IP,IG,IT)
393      WPGV=PVOL*RHOG
394      WDOTPG(IF,IG,IT)=(WPGV-WPV)/(DCYCLE(IP-1))
395      WPGV=WPVG
396      IF(DIAG(2,6HTANK9 )) WRITE (6,6012) IG,IT,IP,IW,IF,
397      1      T,Z,RHOG,RHOLI,PVOL,PRES(IP,IG,IT),PPV,
398      2      WPGV,WDOTPG(IF,IG,IT),DCYCLE(IP-1),WPV
399      C
400      GO TO 90
401      C
402      C      ***** DO FINAL CALCULATIONS FOR SYSTEM 3.
403      C
404      C      ***** FIND MAXIMUM FLOWRATE OF THE PROPELLANT GAS.
405      C      * LOOK UP THE HEX WEIGHT PER FLOWRATE FOR THE HEX DELTA P.

```

```

*****
TANK *****
406 C ***** CALCULATE THE WEIGHT OF THE HEX.
407 C
408 C
409 C DELETE NON-EXISTANT TABLES
410 C
411 C 3010 CONTINUE
412 C
413 WDPGMX := WDOTPG(1,IG,IT)
414 DO 3020 I1 = 2,IF
415 WDPGMX := AMAX1(WDPGMX,WDOTPG(I1,IG,IT))
416 3020 CONTINUE
417 C
418 JX = JX + 1
419 C
420 WDOTX(JX,IG) = WDPGMX
421 UCODE(JX,IG) = HXCODE(JX,IG)
422 C
423 CALL HEATEX(IG, JX, WDOTX(JX,IG), HEXHIT(JX,IG), HEXCIT(JX,IG),
424 1, HEXHOT(JX,IG), HEXCOT(JX,IG), HEXHIP(JX,IG), HEXCIP(JX,IG),
425 2, HEXHOP(JX,IG), HEXCOP(JX,IG), HXMRAT(JX,IG), WDOTH(JX,IG),
426 3, WHXTOT(JX,IG))
427 C
428 WTHXPG(IG) := WHXTOT(JX,IG)
429 C
430 C ***** COMPUTE THE WEIGHT OF GAS GENERATOR PROPELLANT REQUIRED
431 C
432 CALL FINTAB (NTBID(31)+IG)
433 XTAB(2) := SPGTEM(IG,IT)
434 XTAB(3) := SHOTEM(IG,IT)
435 WT = 0.0
436 DO 3030 I1 = 1,IF
437 XTAB(1) := PRES(2*I1-1,IG,IT) + SPDELPG(IG,IT)
438 XTAB(4) := WDOTPG(I1,IG,IT)
439 WT = WT + MIPE(4,XTAB)*DCYCLE(2*I1+1)
440 3030 CONTINUE
441 C
442 WGGPPG(IG,IT) := WT
443 C
444 C ***** LOOK UP GAS GENERATOR SYSTEM WEIGHT.
445 C
446 ATERM = 13.824204 - (0.0117823*SGGPC(IG,IT)) + (1.8632927E-5 *
447 1(SGGPC(IG,IT)**2)) - (1.108423E-8 * (SGGPC(IG,IT)**3))
448 C
449 BTERM = 7.9470262 - (.035636198*SGGPC(IG,IT)) + (6.4684644E-5 *
450 1(SGGPC(IG,IT)**2)) - (3.7946E-8 * (SGGPC(IG,IT)**3))
451 C
452 WGGAPG(IG,IT) = ATERM + BTERM * WDPGMX
453 C
454 C ***** CALCULATE MOTOR HOURE POWER REQUIRED.
455 C ***** NOTE - OVERALL MOTOR PUMP EFFECIENCY FIXED AT 0.5.
456 C
457 CALL RHOLIQ(SITEMP(IG,IT),IG,R)
458 HPMXPG = 144.0*SPDELPG(IG,IT)*WDPGMX/(550.0*0.5*R)
459 C
460 C ***** LOOK UP MOTOR WEIGHT. MOTOR SPEED = CONSTANT = 30000.0
461 C
462 CALL FINTAB (NTBID(33))
463 XTAB(1) := HPMXPG

```

```

***** TANK *****
464      XTAB(2) = 30000.0
465      WMPG(IG,IT) = MIPE(2,XTAB)
466      C
467      C      ***** CALCULATE BATTERY WEIGHT.
468      C
469      WB = 0.0
470      CNST = 144.0*SPDELP(IG,IT)*746.0/(550.0*0.9*R*3600.0)
471      C
472      DO 3040 I1 = 1, IF
473      WB = WB + CNST*WDOTPG(I1,IG,IT)*DCYCLE(2*I1+1)
474      C
475      3040 CONTINUE
476      C
477      WBP(IG,IT) = WB /PDNSTY
478      C
479      C      ***** LOOK UP PUMP WEIGHT.
480      C      ***** NOTE - DELTA PRES. NPSH = 0, AND EFFICIENCY = .7
481      C
482      CALL FINTAB (NTBID(34)+IG)
483      XTAB(1) = .7
484      XTAB(2) = 0.0
485      XTAB(3) = SPDELP(IG,IT)
486      XTAB(4) = WDPGMX
487      WCPPG(IG,IT) = MIPE(4,XTAB)
488      C
489      C      ***** DETERMINE THE PRESSURIZATION SYSTEM WEIGHT.
490      C
491      WTSYPG(IG,IT) = WTHXPG(IG) + WGGAPG(IG,IT) + WMPG(IG,IT) +
492      WBP(IG,IT) + WCPPG(IG,IT)
493      C
494      C      ***** SEPARATE THE GAS GENERATOR PROPELLANT WEIGHT INTO
495      C      ***** H2 AND O2 COMPONENTS
496      C
497      WTGGH2(IG,IT) = WGGPPG(IG,IT)/(SGMRAT(IG,IT) - 1.0)
498      WTGGO2(IG,IT) = WGGPPG(IG,IT) - WTGGH2(IG,IT)
499      C
500      IF(DIAG(2,6)HTANK12) WRITE (6,6015) IG,IT,IP,IW,IF,
501      1 WDPGMX,WTHXPG(IG),WT,WGGPPG(IG,IT),WGGAPG(IG,IT),R,HPMXPG,
502      2 WMPG(IG,IT),CNST,WB,WBP(IG,IT),WCPPG(IG,IT),SPDELP(IG,IT),
503      3 WTSYPG(IG,IT),WTGGH2(IG,IT),SGMRAT(IG,IT),WTGGO2(IG,IT)
504      C
505      GO TO 30
506      C
507      6000 FORMAT ('+',14X,11I5 / (15X,4E15.8))
508      C
509      6005 FORMAT ('+',14X,T23,'-TEMP-',T37,'-WAPS-',T52,'-WP-',T67,'-RHOL',I,
510      1 T82,'-VLIG-',T97,'-TANK-VOL',15X,6E15.8/T23,'-ULL-VOL',T37,'-RHOG-',
511      2 T52,'-PPV-',T67,'-WTOT-',15X,4E15.8)
512      C
513      6010 FORMAT ('+',14X,5I5/(15X,7E15.8))
514      C
515      6012 FORMAT ('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
516      1 'IF=',I3/15X,T21,'TEMP-GAS',T37,'Z-GAS',T52,'RHOG',T67,'RHOLIG',
517      2 T81,'ULL-VOL',T97,'PRESSURE',T112,'-PPV-',15X,7E15.8/15X,T23,
518      3 'WPVG',T37,'WDOTPG',T52,'DCYCLE',T67,'-WPV-',15X,4E15.8)
519      C
520      6015 FORMAT ('+',14X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
521      1 'IF=',I3/15X,T23,'WDPGMX',T37,'WTHXPG',T52,'- WT-',T67,'WGGPPG',

```

***** TANK *****

```

522      2 T82,'WGGAPG',T97,'- R -',T112,'HMPXPG',/15X,TE15.8/15X,T23,'HMPG',
523      3 T37,'CNST',T52,'- WB -',T67,'HBPBG',T82,'WCPPG',T97,'SPDELP',/15X,
524      4 6E15.8/15X,T23,'WTSYPG',T37,'WTGGH2',T52,'SGHRAT',T67,'WTGG02',
525      5 /15X,4E15.8)
526      C
527      6020 FORMAT(/20X,20(' '), ' BURN NUMBER =',I4,5X,'PRESS.SYS.NO. =',I4,
528      1 1X,20(' ')/)
529      C
530      6030 FORMAT(/20X,20(' '), ' COAST NUMBER =',I4,5X,'PRESS.SYS.NO. =',I4,
531      1 1X,20(' ')/)
532      C
533      6031 FORMAT(/18X,20(' '), ' FINAL ENGINE SHUTDOWN PROPELLANT TANK CONDI
534      ITIONS ',20(' ')/)
535      C
536      6033 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
537      1 'IF=',I3/15X,T23,'RHOP',T34,'INT.ENERGY',T54,'WP',T69,'WPV',T84,
538      2 'WPT',T96,'PRESSURE',/15X,6E15.8)
539      6035 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
540      1 'IF=',I3/15X,T23,'SVWT',T37,'WTOT',T52,'PPV',T68,'WP',T83,'PHE',
541      2 'T96,'PRESSURE',/15X,6E15.8/15X,T23,'RHOP',T37,'RATIO',T51,
542      3 'INT.ENERGY',/15X,3E15.8)
543      6040 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
544      1 'IF=',I3,
545      2 T66,'WTTOT',T81,'WDOTJ',T94,'INT.ENERGY',T113,'WTOT',/15X,7E15.8,
546      3 /15X,T22,'DCYCLE',T38,'-E-',/15X,2E15.8)
547      6050 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
548      1 'IF=',I3/15X,T23,'WOUT',T37,'WTOT',T53,'WPT',T66,'INT.ENERGY',
549      2 T82,'WTTOT',T97,'PWTOT',T112,'-E-',/15X,7E15.8/15X,T23,'SVOL',
550      3 T37,'RHOP',/15X,2E15.8)
551      6060 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
552      1 'IF=',I3/15X,T23,'RHOP',T34,'INT.ENERGY',T54,'CVP',T69,'CVH',T82,
553      2 'RATIO',T99,'PPV',/15X,6E15.8/T24,'PHE',T36,'PRESSURE',T54,'WP',
554      3 T69,'WPV',T82,'RHOG',/15X,5E15.8)
555      6061 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
556      1 'IF=',I3/15X,T23,'RHOP',T34,'INT.ENERGY',T51,'RATIO',T70,'PPV',
557      2 T80,'PRESSURE',/15X,5E15.8)
558      6070 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
559      1 'IF=',I3/15X,T21,'WYSUM',T36,'TFINAL',T53,'-2-',T66,'SHVTOT',
560      2 T83,'WGR',T97,'WLR',/15X,6E15.8)
561      6080 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
562      1 'IF=',I3/15X,T22,'TSAT',T37,'Z-GAS',T51,'RHO-GAS',T66,'RHO-LIQ',
563      2 T81,'ULL-VOL',T95,'LIO-VOL',T112,'-WP-',/15X,7E15.8/T23,'-WPV-',
564      3 T34,'OP.PRESSURE',T52,'PPV',T67,'HE-TEMP',T81,'Z-HELIUM',T97,
565      4 '-WH-',T112,'-WHE-',/15X,7E15.8/T23,'DCYCLE',T37,'WDOHE',T52,
566      5 'PRESSURE',T65,'HE-PRESSURE',/15X,4E15.8)
567      6090 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
568      1 'IF=',I3/15X,T22,'SVWT',T38,'- WP -',T53,'RHO',T64,'INT.ENERGY',
569      2 15X,5E15.8)
570      6100 FORMAT(' ',4X,'IG=',I3,2X,'IT=',I3,2X,'IP=',I3,2X,'IW=',I3,2X,
571      1 'IF=',I3/15X,T22,'SUMWDH',T38,'WHETOT',T51,'WPGTOT',T67,'WPTOT',
572      2 T83,'PWTOT',/15X,5E15.8)
573      C
574      7000 FORMAT(/T42, '*** TANK AND VENT PARAMETER CALCULATIONS ***'/)
575      7001 FORMAT(T20, '*** INITIAL TANK CONDITIONS ***'/)
576      7002 FORMAT(T5, 'FLUID CONSIDERED = ',2A6,T43, 'FLUID TEMPERATURE
577      1,F8.2,T81, 'TANK INITIAL PRESSURE = ',F8.2/T5, 'WGT.OF LIQ. PROP.
578      2 = ',F8.2,T43, 'WGT. PROP. VAPOR = ',F8.3,
579      3
580      T81, 'WGT. LIQ. + VAPOR

```



```

***** TANK *****
580      4 = 'F8.2/T5,'WGT. HELIUM IN VAPOR = 'F8.2,T43,'TOTAL FLUIDS IN
581      5TANK = 'F8.2,T81,'VOL. OF LIQUID FLUID = 'F8.2/T5,'PART.PRES.P
582      6PROP.VAPOR = 'F8.3,T43,'PART.PRES.HELIIUM GAS = 'F8.3,T81,'ULLAGE
583      7 VOLUME IN TANK = 'F8.2/T5,'TANK VOLUME = 'F8.2,T43,'E
584      8FF. TANK DENSITY = 'F8.3,T81,'EFF. INTERNAL ENERGY = 'E15.8
585      9)
586      7003 FORMAT(T20,'*** PRE- OR NON-VENT CONDITIONS ***')
587      7004 FORMAT(T5,'FLUID CONSIDERED = '2A6,T43,'FLUID TEMPERATURE = '
588      1,'F8.2,T81,'COAST DURATION = SEC. = 'F8.0/T5,'WGT.OF LIQ. PROP.
589      2 = 'F8.3,T43,'WGT. PROP. VAPOR = 'F8.3,T81,'WGT.HELIIUM IN
590      3VAPOR = 'F8.3/T5,'PART.PRES.PROP.VAPOR = 'F8.3,T43,'PART.PRES.
591      4HELIIUM GAS = 'F8.3,T81,'CURRENT TANK PRESSURE = 'F8.3/T5,'EFF.I
592      5INTERNAL ENERGY = 'E15.8)
593      7005 FORMAT(/T20,'*** POST VENT CONDITIONS ***')
594      7006 FORMAT(T5,'TANK VENT PRESSURE = 'F8.2,T43,'WGT.VENTED FLUIDS
595      1 = 'F8.2,T81,'WGT.OF LIQ.IN TANK = 'F8.2/T5,'WGT.VAPOR IN TA
596      2NK = 'F8.3,T43,'WGT.HELIIUM IN VAPOR = 'F8.3,T81,'TOTAL FLUI
597      3DS IN TANK = 'F8.2/T5,'PART.PRES.PROP.VAPOR = 'F8.3,T43,'PART.
598      4PRES.HELIIUM GAS = 'F8.3,T81,'VENTED TANK PRESSURE = 'F8.3/T5,
599      5 'EFF.INTERNAL ENERGY = 'E15.8)
600      7007 FORMAT(T20,'*** COMPUTE ENERGY BALANCE FOR BURN ***')
601      7008 FORMAT(T5,'FLUID CONSIDERED = '2A6,T43,'BURN DURATION = SEC. = '
602      1,'F8.0,T81,'FLOWRATE FOR THRUST = 'F8.3/T5,'THRUST PROP.REMAININ
603      2G = 'F8.2,T43,'PROPELLANT IN TANK = 'F8.2,T81,'EFF. INTERNAL
604      3ENERGY = 'E15.8/T5,'EFF. TANK ENERGY = 'E15.8,
605      4 T81,'TOTAL FLOWRATE = 'F8.3)
606      7009 FORMAT(/T20,'*** COMPUTE RESULTING TANK CONDITIONS ***')
607      7010 FORMAT(T5,'PROPELLANT WITHDRAWN = 'F8.3,T43,'TOTAL FLUIDS IN TAN
608      1K = 'F8.2,T81,'PROPELLANT LIQ.+VAP. = 'F8.2/T5,'THRUST PROP.RE
609      2MAINING = 'F8.2,T43,'NEW EFF. TANK DENSITY = 'F8.4,T81,'PART.PRE
610      3S.PROP.VAPOR = 'F8.3/T5,'NEW INTERNAL ENERGY = 'E15.8)
611      7011 FORMAT(/T20,'*** COMPUTE PRESSURANT NEEDED FOR THIS BURN ***')
612      7012 FORMAT(T5,'TANK LIQ. TEMPERATURE = 'F8.2,T43,'STORED HELIUM TEMP.
613      1 = 'F8.2,T81,'NEW TANK ULLAGE VOL. = 'F8.3/T5,'NEW PROP. LIQ.
614      2 VOLUME = 'F8.2,T43,'PROP. LIQ. REMAINING = 'F8.2,T81,'WGT. OF
615      3PROP. VAPOR = 'F8.4/T5,'HELIIUM PART.PRESSURE = 'F8.3,T43,'TOT
616      4AL PRES. *PPV+PHE* = 'F8.3,T81,'NOM. OPERATING PRES. = 'F8.3/T5
617      5,'HELIIUM FLOW RATE = 'E10.4,T43,'WEIGHT OF HELIUM USED = '
618      6 E10.4,T81,'NEW TANK PRESSURE = 'F8.3/T5,'TOTAL HELIUM CONSUM
619      7ED = 'F8.3)
620      7013 FORMAT(T20,'*** COMPUTE FINAL TANK CONDITIONS ***')
621      7014 FORMAT(/T5,'FINAL TANK TEMP. = 'F8.3,T43,'TOTAL VENTED GAS.W
622      1GT. = 'F8.3,T81,'WGT. OF GAS RESIDUALS = 'F8.3/T5,'WGT. OF LIQ.R
623      2ESIDUALS = 'F8.3)
624      7015 FORMAT(/T20,'*** COMPUTE PRESSURIZATION SYSTEM WEIGHT ***')
625      7016 FORMAT(T5,'TOTAL HELIUM GAS REQD = 'F8.3,T43,'WGT.PRESSURANT SYST
626      1EM = 'F8.3)
627      7017 FORMAT(/T42,'*** TANK AND VENT PARAMETER CALCULATIONS - CONTO. ***
628      1')
629      C
630      C
631      END

```

PROCEDURE DEFINITION PROCESSOR - TANKWT

```

1  TANKWT* PROC
2  C
3      PARAMETER NIW=10,N2W=2
4  C
5      COMMON /TANKWT/ IWOP,NOSHAP,JTKTYP(NIW),JFLTP(NIW),XD(NIW),YD(NIW)
6      I ,ZD(NIW),TAR(NIW),TVL(NIW),THD(N2W),TOTVL(N2W)
7  C
8      DIMENSION A1(N2W),A2(N2W),A3(N2W),V1(N2W),V2(N2W),V3(N2W),VMX(N2W)
9  C
10     EQUIVALENCE (TOTVL,VMX),(TAR,A1),(TAR(3),A2),(TAR(5),A3),(TVL,V1),
11     I (TVL(3),V2),(TVL(5),V3)
12  C
13     END

```

***** FUNCTION TBØIL

```
1      FUNCTION TBØIL(I)
2      DIMENSION TC(17)
3      DATA(TC(N),N=1,17)/277.85,59.8,259.13,59.8,260.,343.2,387.2,
4      343.2,387.2,521.8,259.13,730.,776.4,1094.,749.5,1155.,9.37/
5      TBØIL=TC(I)
6      RETURN
7      END
```

SUBROUTINE TCØND

```

1      SUBROUTINE TCOND(TH,TC,NBAR,THKIN,INTYPE,QCOND)
2      C
3      C
4      C THIS SUBROUTINE COMPUTES THE THERMAL CONDUCTIVITY FOR NINE
5      C INSULATION MATERIALS. REF., LMSC-A964947-VOL.II, LMSC A981608,
6      C WITH MICROSPHERE EQNS. PER R.PARMLEY. EMITTANCE VALUES ARE FROM
7      C LMSC A903316 (NASA CR-72605).
8      C
9      C REAL NSHLD,NBAR
10     C
11     DELT = TH-TC
12     TMEAN = (TH+TC)/2.0
13     SUMT = TH+TC
14     SUMSQ = (TH**2)+(TC**2)
15     TMPRI = TC/TH
16     TMPR2 = TMPRI*TMPRI
17     TH3 = TH*TH*TH
18     NSHLD = THKIN*NBAR
19     THKFT = THKIN/12.0
20     THETA1 = (1.0+TMPRI)
21     THETA2 = (1.0+TMPR2)
22     SIGMA = 0.1713E-08
23     C
24     GO TO (10,20,30,40,50,60,70,80,90),INTYPE
25     C
26     C * FOR DOUBLE ALUMINIZED MYLAR-SILK NET
27     C
28     10 SCNST = 2.22E-09
29     EMIT1 = 4.40E-04 * (TMEAN**0.667)
30     DEMIT = ((2.0/EMIT1)-1.0)
31     GO TO 22
32     C
33     C * FOR DOUBLE GOLDIZED MYLAR-SILK NET
34     C
35     20 SCNST = 2.22E-09
36     EMIT2 = 8.76E-04 * (TMEAN**0.509)
37     DEMIT = ((2.0/EMIT2)-1.0)
38     22 SCOND = SCNST*NBAR*TMEAN
39     RNUM = SIGMA*SUMSQ*SUMT*THKFT
40     RDEN = (NSHLD-1.0) * DEMIT
41     RCOND = RNUM/RDEN
42     QCOND = (SCOND+RCOND) *(DELT/THKFT)
43     RETURN
44     C
45     C * FOR DOUBLE ALUMINIZED MYLAR-TISSUE GLASS
46     C
47     30 SCNST = 7.00E-12
48     EMIT1 = 4.40E-04 * (TMEAN**0.667)
49     DEMIT = ((2.0/EMIT1)-1.0)
50     GO TO 42
51     C
52     C * FOR CRINKLED DOUBLE ALUMINIZED MYLAR-TISSUE GLASS
53     C
54     40 SCNST = 8.80E-12
55     EMIT1 = 4.90E-04 * (TMEAN**0.67)
56     DEMIT = ((2.0/EMIT1)-1.0)
57     42 SCOND = SCNST*(NBAR**2)*TMEAN
58     RNUM = 1.7*SIGMA*SUMSQ*SUMT*THKFT

```

```

***** TCOND *****
58      RDEN = (NSHLD-1.0) * DEMIT
59      RCOND = RNUM/RDEN
60      QCOND = (SCOND+RCOND) * (DELT/THKFT)
61      RETURN
62      C *
63      C * FOR NRC-2 CRINKLED SINGLE ALUMINIZED MYLAR
64      C *
65      50 SCNST = 2.00E-10
66      EMITA = 4.90E-04 * (TMEAN**0.67)
67      EMITB = 5.58E-03 * (TMEAN**0.667)
68      GO TO 62
69      C *
70      C * FOR SUPERFLOC
71      C *
72      60 SCNST = 15.40E-11
73      EMITA = 4.40E-04 * (TMEAN**0.667)
74      EMITB = 4.10E-01
75      62 DEMIT = ((1.0/EMITA)+(1.0/EMITB)-1.0)
76      SCOND = SCNST*(NBAR**2)*TMEAN
77      RNUM = SIGMA*SUMSQ*SUMT*THKFT
78      RDEN = (NSHLD-1.0)*DEMIT
79      RCOND = RNUM/RDEN
80      QCOND = (SCOND+RCOND)*(DELT/THKFT)
81      RETURN
82      C *
83      C * FOR MICROSPHERES (104 TO 135 MICRONS)
84      C *
85      70 RCNST = 1.56E-13
86      RCOND = RCNST*TH3*THETA1*THETA2
87      QCOND = RCOND * (DELT/THKFT)
88      RETURN
89      C *
90      C * FOR POLYURETHANE FOAM
91      C *
92      80 PKSUBE = 1.1295E-03 + (3.481E-05 * TMEAN)
93      QCOND = PKSUBE * (DELT/THKFT)
94      RETURN
95      C *
96      C * FOR FIBERGLASS BATTING - HELIUM PURGED
97      C *
98      90 FKSUBE = 1.3836E-03 * (TMEAN**0.662)
99      QCOND = FKSUBE * (DELT/THKFT)
100     RETURN
101     C *
102     END

```

***** SUBROUTINE TEL

```

1      C      * * * * *
2      C      * ROUTINE NAME - TABLE EVALUATION OR LOOKUP *
3      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
4      C      * PROGRAMMER   - R. BOLLINGER 1949 102 26933 *
5      C      * DATE CODED   - 2/25/70 *
6      C      * * * * *
7      C
8      C      SUBROUTINE TEL(X,Y)
9      C
10     C      ***** EXPLANATION OF THE CALLING SEQUENCE
11     C
12     C      X - VALUE OF THE INDEPENDENT VARIABLE TO BE
13     C      INTERPOLATED ON.
14     C
15     C      Y - VALUE OF THE DEPENDENT VARIABLE WHICH IS THE
16     C      INTERPOLATION RESULTS.
17     C
18     C      INCLUDE CIOUNT
19     C      INCLUDE CKEYS
20     C      INCLUDE CTAB
21     C
22     C      6000 FORMAT ('010X1*** INTERPOLATION ERROR - OVERFLOW OR NV WRONG NV#',
23     C      ' I3, ' ***')
24     C      6100 FORMAT ('1010X1*** INDEP. VAR. = F10.4, ' OUT OF TABLE RANGE (I
25     C      ' F10.4, ' TO' F10.4, ' ) TABLE NO. I4, ' NO. SUBTABLES I3, ' ***')
26     C
27     C
28     C      IF (TYPE.EQ.1) GO TO 30
29     C
30     C      ***** EVALUATE THE COEFFICIENT POLYNOMIAL FOR Y.
31     C
32     C      Y = XTAB(I)
33     C      DO 20 I1 = 2,NV
34     C      Y = XTAB(I1) + X*Y
35     C      20 CONTINUE
36     C      RETURN
37     C
38     C      ***** EVALUATE THE TABLE LOOKUP FOR Y.
39     C
40     C      30 IF (XTAB(I) .LE. XTAB(NV)) GO TO 33
41     C      MAKE INDEPENDENT VAR. NON-DECREASING
42     C      NV1 = NV + 1
43     C      NVD2 = NV / 2
44     C      DO 32 I=1,NVD2
45     C      PIVOT = XTAB(NV1-I)
46     C      XTAB(NV1-I) = XTAB(I)
47     C      XTAB(I) = PIVOT
48     C      PIVOT = YTAB(NV1-I)
49     C      YTAB(NV1-I) = YTAB(I)
50     C      32 YTAB(I) = PIVOT
51     C      33 IF (KEY1 .LT. 1612) GO TO 34
52     C      TEST FOR INDEP. VAR. OUT OF RANGE
53     C      IF (X .GE. XTAB(1) .AND. X .LE. XTAB(NV)) GO TO 34
54     C      WRITE (IOT,6100) X,XTAB(1),XTAB(NV),JTABID,NLTBL
55     C      IF (KEY1 .NE. 1612) GO TO 34
56     C      KEY2 = KEY1
57     C      RETURN

```

```

***** TEL *****
58      34 Y= YLGINT(XTAB,YTAB,NV,X,NIP,S40)
59      RETURN
60      40 WRITE (107,6000) NV
61      CALL EXIT
62      C
63      END

```

SUBROUTINE TEMP

```

1 SUBROUTINE TEMP(T)
2 C
3 C.... ROUTINE TO CONVERT NBS-55 TO IPTS-68 IN THE RANGE BELOW NBP-OXYGEN
4 C.... (90.18 NBS-55 OR 90.188 IPTS-68) BY INTERPOLATION FROM TABULATED VALUES
5 C.... GIVEN IN METROLOGIA 5, 47(1967)
6 C
7 C.... ROUTINE TO CONVERT IPTS-48 TO IPTS-68 BY METHOD GIVEN BY
8 C.... THOMAS B. DOUGLAS (JOURNAL OF RESEARCH NBS VOL 73A NO5 SEPT-OCT 1969)
9 C.... IN THE RANGE 90.18 TO 10000 K.
10 C
11 DIMENSION CHNG(38)
12 C
13 DATA (CHNG(I),I=1,38) /8,900,7,100,5,200,3,400,1,700,0,900,-0,800,
14 2 -1,400,-1,500,-1,200,-0,700,-0,100,0,500,0,900,1,100,0,900,0,300,
15 3 -0,600,-1,700,-3,000,-4,300,-5,600,-6,800,-7,800,-8,600,-9,000,
16 4 -9,000,-8,600,-7,700,-6,400,-4,900,-2,900,-0,500,2,200,4,900,
17 5 7,400,9,600,11,100/
18 IF((T.LT.273.150100).AND.(T.GT.273.149900))RETURN
19 IF(T.GT.90.1800)GO TO 2
20 IT=T
21 TI=IT
22 IT=IT-53
23 DELT=(CHNG(IT)+((CHNG(IT+1)-CHNG(IT))*(T-TI))/1000.00
24 T=T+DELT
25 RETURN
26 2 IF(T.GT.273.1500)GO TO 3
27 C=T-273.15
28 CALL WFIN(T,W)
29 TOP=1.E0+3.984517E-3*C-5.855019E-7*C**2+4.35717E-12*(100.00-C)*
30 1 C**3-W
31 TOP=TOP*250.97
32 BOT=1.00-2.9389E-4*C+4.3741E-9*(75.00-C)*C**2
33 DELT=TOP/BOT
34 T=T+DELT
35 RETURN
36 3 IF(T.GT.903.8900)GO TO 4
37 C=T-273.15
38 TOP=4.904E-7*C*(C-100.00)
39 BOT=1.00-2.939E-4*C
40 PT=0.045
41 PT=PT*(C/100.00)
42 PT=PT*(C/100.00-1.00)
43 PT=PT*(C/419.5800-1.00)
44 PT=PT*(C/630.7400-1.00)
45 DELT=TOP/BOT
46 DELT=DELT+PT
47 T=T+DELT
48 RETURN
49 4 IF(T.GT.1337.5800)GO TO 5
50 C=T-273.15
51 TOP=-1.3145+1.5016E-3*C+1.5625E-6*C**2
52 BOT=1.000+4.101E-4*C
53 DELT=TOP/BOT
54 T=T+DELT
55 RETURN
56 5 IF(T.GT.10273.1500)WRITE(6,100)T
57 100 FORMAT(1 TEMP TO BE CONVERTED EXCEEDS 10000 C INPUT WAS,P20,10)

```



```

***** TEL *****
58      34 Y = YLGINT(XTAB,YTAB,NV,X,NJP,340)
59      RETURN
60      40 WRITE (10T,6000) NV
61      CALL EXIT
62      C
63      END

```

```

*****  TEMP  *****
58      E= EXP(-22135.00/T)
59      DELT=5.56E-4*T+3.84E-7*(1.00-E)*T**2
60      T = T + DELT
61      RETURN
62      END

```

***** SUBROUTINE THETAB

```
1      SUBROUTINE THETAB(DB,CPB,D1B,D2B,SHI)
2      C
3      C CALCULATES SPECIFIC HEAT INPUT (THETA) FROM EGN OF STATE
4      C IN BRITISH UNITS
5      C MUST FOLLOW CALLS OF PROPB,CPVTDB,D1DDB, AND D1DDB TO DEFINE
6      C DB,DPB,D1B,+D2B
7      C INPUT AND OUTPUT PARAMETERS ARE IN BRITISH UNITS
8      C SHI=DB*CPB*(D1B/D2B)
9      RETURN
10     END
```

***** SUBROUTINE THKWTG

```

1  C
2  SUBROUTINE THKWTG (KFLG,IGAS,ITYPE,MTYPE,TEMP,PU,HD,AREA,HT,RAD,
3  ,RADI,TWEGHT,KMT)
4  C
5  C      *****
6  C      * THIS ROUTINE CALCULATES WALL *
7  C      * THICKNESS AND WEIGHT (IF NOT *
8  C      * SPECIFIED WEIGHT IS *
9  C      * MINIMIZED) *
10 C      * DATE CODED JULY 1972 *
11 C      * PROGRAMMER J. MCKAY , D19-43 *
12 C      * BLDG 201 X45178 *
13 C      *****
14 C
15 INCLUDE 'CHATRL.LIST'
16 INCLUDE 'TABLOK'
17 C
18 DIMENSION ER(5),IM(3),RK(5,2),WT(3)
19 C
20 DATA IM/2,1,5/
21 DATA ER/1,1.25,1.5,1.75,2./
22 DATA (RK(I,1),I=1,5) / .50,.605,.80,.925,1.13 /
23 DATA (RK(I,2),I=1,5) / .67,.745,.84,.955,1.20 /
24 C
25 C      KFLG = 1 FOR COMPLETE ELLIPSOID
26 C      = 2 ELLIPSOID CONNECTED TO OTHER SHAPE
27 C      IGAS = 1 FOR O2 , = 2 FOR H2
28 C      ITYPE= TANK SHAPE 1=CYL,2=FRUS,OF CONE,3=ELLIPSOID
29 C      4=CYD.=ELLIPSOID
30 C      MTYPE= TANK MATERIAL TYPE (IF NOT INPUT PROGRAM WILL
31 C      SELECT LIGHTEST)
32 C
33 C      FIND FLUID DENSITY
34 CALL FDNSTY (IGAS,TEMP,PU,RHOF)
35 C      TANK PRESSURE
36 P = PU + 0.0229*HD*RHOF
37 JKM = 0
38 IF (P .GE. 1000.) JKM = 5
39 IF (P .GE. 3000.) JKM = 10
40 ISTD = 1
41 IF (MTYPE .EQ. 0) ISTD = 3
42 KMT = IM
43 C
44 DO 200 I=1,ISTD
45 IMTR = IM(I)
46 IF (MTYPE .EQ. 0) GO TO 100
47 KMT = MTYPE
48 IMTR = MTYPE
49 100 CALL FINTAB (NTBID(25)+IMTR)
50 C      LOOKUP FTU FOR THE PARTICULAR MATERIAL (IMTR)
51 FTU = MIPE(I,TEMP)
52 C      CALCULATE THICKNESS FOR THE GIVEN SHAPE
53 GO TO (110,120,130,110),ITYPE
54 C      CYLINDER OR CYLINDER - ELLIPSOID
55 110 THK = 2.0*P*RAD / FTU
56 GO TO 180
57 C      FRUSTRUM OF CONE

```

```

***** THKWTG *****
58 C CALC. CONE HALF ANGLE
59 120 COSALP = HT / (SQRT(HT*HT + (RAD1-RAD)**2))
60 THK = 2.0*P*AMAX1(RAD,RAD1) / (COSALP*FTU)
61 GO TO 180
62 C ELLIPSOID (SPHERICAL)
63 C CALC. ELLIPSE RATIO
64 130 ERATO = AMAX1(HT,RAD) / AMIN1(HT,RAD)
65 C INTERPOLATE ON ERATO TO FIND K
66 DO 140 J=1,5
67 J1 = J
68 IF (ERATO - ER(J)) 140,160,150
69 140 CONTINUE
70 C
71 150 YK = RK(J1-1,KFLG)+(ERATO-ER(J1-1))*(RK(J1,KFLG)-RK(J1-1,KFLG))
72 / (ER(J1)-ER(J1-1))
73 GO TO 170
74 160 YK = RK(J1,KFLG)
75 170 THK = (ERATO/2.+YK)*P*AMAX1(HT,RAD) / (2.0*FTU)
76 C CONVERT THICKNESS FROM FEET TO INCHES
77 180 THK = 12.0 * THK
78 C CHECK FOR LESS THAN ALLOWABLE THICKNESS
79 IF (THK .LT. MINTHK(IMTR+JKM)) THK = MINTHK(IMTR+JKM)
80 C CALC. WEIGHT FOR EACH MATERIAL
81 WT(I) = 0.1125 * RHOL(IMTR) * THK * AREA
82 C SAVE MIN WEIGHT AND MATERIAL TYPE
83 IF (WT(I) .GE. WT) GO TO 200
84 WT = WT(I)
85 KMT = IMTR
86 200 CONTINUE
87 C SET WEIGHT EQUAL TO LIGHTEST
88 TWEGHT = WT
89 RETURN
90 END

```

FUNCTION VFUNC (TKGEØM)

```

1  C
2  FUNCTION VFUNC (I)
3  C
4  C      * * * * *
5  C      * ROUTINE INTERPRETS INPUT DIMENSIONS *
6  C      * TO CALL GEOMETRY PROGRAMS *
7  C      * DATE CODED JULY 1972 *
8  C      * PROGRAMMER J. MCKAY , D19-43 *
9  C      * BLDG 201 X45178 *
10 C      * * * * *
11 C
12 C INCLUDE TANKWT,LIST
13 C
14 C JMP = IABS (JTKTYP(I))
15 C
16 C      VOLUME CALCULATION ROUTINES
17 C      GO TO (100,110,120,130),JMP
18 C      CYLINDER
19 C      100 VFUNC = CYLNDR (YD(I),XD(I))
20 C      IF (IWOP .EQ. 3) VFUNC = 0.
21 C      GO TO 500
22 C      FRUSTRUM OF CONE
23 C      110 VFUNC = FRCONC (YD(I),XD(I),ZD(I))
24 C      GO TO 500
25 C      ELLIPSOID
26 C      120 VFUNC = HSPHER (XD(I),YD(I))
27 C      GO TO 500
28 C      CYLINDER LESS ELLIPSOID
29 C      130 VFUNC = CYLSPH (XD(I),YD(I))
30 C      GO TO 500
31 C
32 C      AREA CALCULATION ROUTINES
33 C      ENTRY AFUNC (I)
34 C
35 C      JMP = IABS (JTKTYP(I))
36 C      GO TO (200,210,220,230),JMP
37 C      CYLINDER
38 C      200 VFUNC = ARACYL (YD(I),XD(I))
39 C      GO TO 250
40 C      FRUSTRUM OF CONE
41 C      210 VFUNC = AREAFR (YD(I),XD(I),ZD(I))
42 C      GO TO 250
43 C      ELLIPSOID
44 C      220 VFUNC = ARSPHR (XD(I),YD(I))
45 C      GO TO 250
46 C      CYLINDRICAL + ELLIPSOIDAL SHAPE
47 C      230 VFUNC = ARACYL (YD(I),XD(I))
48 C      SHOULD AREA OF ELLIPSOID BE INCLUDED
49 C      IF (JFLTYP(I) .LT. 0) GO TO 500
50 C      YES INCLUDE IT
51 C      VFUNC = VFUNC + ARSPHR (XD(I),YD(I))
52 C      GO TO 500
53 C      250 IF (JFLTYP(I) .LT. 0) VFUNC = 0.
54 C      GO TO 500
55 C
56 C      HEAD CALCULATION ROUTINES
57 C      ENTRY HFUNC (I,PVOL)

```

```

***** TKGEOM *****
58 C
59     JMP = IABS (JTKTYP(I))
60     GO TO (300,310,340,370),JMP
61 C     CYLINDER
62     300 CALL CYLHED (PVOL,YD(I),HD)
63     VFUNC = HD
64     GO TO 500
65 C     FRUSTRUM OF CONE
66     310 PV = PVOL
67     IF (JTKTYP(I) .GT. 0) GO TO 320
68 C     INVERTED FRUST. OF CONE
69     PV = TVL(I) - PV
70     320 CALL FRHEAD (PV,YD(I),ZD(I),XD(I),HD)
71     IF (JTKTYP(I) .GT. 0) GO TO 330
72     HD = XD(I) - HD
73     330 VFUNC = HD
74     GO TO 500
75 C     ELLIPSOID
76     340 PV = PVOL
77     IF (JTKTYP(I) .GT. 0) GO TO 350
78 C     INVERTED ELLIPSOID
79     PV = TVL(I) - PV
80     350 CALL ELIPSG (PV,XD(I),YD(I),HD)
81     IF (JTKTYP(I) .GT. 0) GO TO 360
82     HD = XD(I) - HD
83     360 VFUNC = HD
84     GO TO 500
85 C     HEAD BETWEEN CYLINDER AND ELLIPSOID
86     370 CALL CYNSPH (PVOL,XD(I),YD(I),HD)
87     VFUNC = HD
88     500 CONTINUE
89     RETURN
90     END

```

***** FUNCTION TMELT

```

1      FUNCTION TMELT(P,K)
2      IF(K.EQ.2)GO TO 1
3      C  = 1.769
4      PO = 2637.2
5      PT = 0.00150
6      TT = 54.3507
7      X  = 1.0/C
8      T  = ((TT**C) * ((P/PO)-(PT/PO)+1.0))**X
9      CALL TEMP(T)
10     TMELT=T
11     RETURN
12     I CONTINUE
13     A  = -1979.0825
14     B  = 0.9263018
15     C  = 1.0/1.795
16     T=((P-A)/B)**C
17     TMELT=T
18     RETURN
19     END

```


***** FUNCTION TMELT B

```
1      FUNCTION TMELTB(PB,K)
2      P = PB * 6.8947572E+3/1.01325E+5
3      T=TMELT(P,K)
4      TMELTB = T * 1.8
5      RETURN
6      END
```

SUBROUTINE TNKWT

```

1 C
2 SUBROUTINE TNKWT (ISW,FLDVOL,PCULLG,DIAM,TKTEMP,TKPRES,MTYPE,
3 TNKVL,WTOFTK,TOTARA,HC)
4
5 C
6 C
7 C
8 C
9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C

```

ROUTINE CONTROLS THE CALCULATION OF
 TANK VOL., AREAS, HEAD, WALL THICK-
 NESS AND WEIGHTS.
 FOR BOTH OXYGEN AND HYDROGEN TANKS
 DATE CODED JULY 1972
 PROGRAMMER J. MCKAY, D19-43
 BLDG 201 X45178

INCLUDE CONST,LIST
 INCLUDE TANKWT,LIST

DIMENSION DIAM(N2W),FLDVOL(N2W),MTYPE(N2W),PCULLG(N2W),RMAX(N2W),
 TKPRES(N2W),TKTEMP(N2W),TNKVL(N2W),TOTARA(N2W),WTOFTK(N2W)
 HC(N2W)

DO 100 I=1,2
 RMAX(I) = DIAM(I) / 2.
 TNKVL(I) = FLDVOL(I) / ((1.0-PCULLG(I))/100.)
 100 TOTVL(I) = 0.

ISW = 1 STORAGE TANKS
 ISW = 2 ACCUMULATOR TANKS
 IF (ISW.EQ. 2) GO TO 110
 IF (IWOP.GT. 1) GO TO 200
 CALCULATE MAX. VOLUME (SPHERE)

110 DO 190 I=1,2
 VMX(I) = SPHERE (RMAX(I),RMAX(I))
 IF (VMX(I).LE. TNKVL(I)) GO TO 130
 CALC. NEW RADIUS OF SPHERE
 RMAX(I) = (FLDVOL(I) / (2.*PI203))**.5
 DIAM(I) = 2.0*RMAX(I)
 120 V2(I) = 0.
 V1(I) = TNKVL(I) / 2.
 V3(I) = V1(I)
 HC(I) = 0.
 GO TO 140

130 IF (VMX(I).EQ. TNKVL(I)) GO TO 120
 REQUIRED VOL. GREATER THAN MAX SPHERE
 CALC. VOL OF ADDED CYLINDRICAL SECTION
 V2(I) = TNKVL(I) - VMX(I)
 CALC. HEIGHT OF CYLINDER
 HC(I) = V2(I) / (PI*RMAX(I)**2)
 V1(I) = VMX(I) / 2.
 V3(I) = V1(I)

140 VMX(I) = TNKVL(I)
 CALCULATE HEAD FOR OPTION IWOP = 1
 IF (FLDVOL(I).GT. V1(I)) GO TO 150
 HEAD IN LOWER HEMIS
 CALL SPHSEG (FLDVOL(I),RMAX(I),THD(I))
 GO TO 170

150 THD(I) = RMAX(I)
 IF (FLDVOL(I).GT. V1(I)+V2(I)) GO TO 160

```

***** TNKMTA *****
58 C HEAD IN CYLINDER
59 PVOL = FLDVOL(I) - VI(I)
60 CALL CYLHED (PVOL, RMAX(I), HCI)
61 THD(I) = THD(I) + HCI
62 GO TO 170
63 160 THD(I) = THD(I) + HC(I)
64 PVOL = TNKVL(I) - FLDVOL(I)
65 C HEAD IN UPPER HEMIS
66 CALL SPHSEG (PVOL, RMAX(I), HCI)
67 THD(I) = THD(I) + RMAX(I) - HCI
68 C FIND AREAS
69 170 A1(I) = ARSPHR (RMAX(I), RMAX(I))
70 A3(I) = A1(I)
71 A2(I) = ARACYL (RMAX(I), HC(I))
72 C TOTAL AREA
73 TOTARA(I) = A1(I) + A2(I) + A3(I)
74 C CALCULATE TANK WEIGHT (IWOP=1)
75 KFLG = 1
76 WTG1 = 0.
77 IF (HC(I) .EQ. 0.) GO TO 180
78 KFLG = 2
79 C CALC. WEIGHT OF CYLINDRICAL SECTION (IF NECESSARY)
80 CALL THKWTG (KFLG, I, 1, MTYPE(I), TKTEMP(I), TKPRES(I), THD(I), A2(I),
81 I HC(I), RMAX(I), RMAX(I), WTG1, MFLG1)
82 C CALC. WEIGHT OF HEMISPHERICAL ENDS
83 180 CALL THKWTG (KFLG, I, 3, MTYPE(I), TKTEMP(I), TKPRES(I), THD(I),
84 I 2.0*A1(I), RMAX(I), RMAX(I), HC, WTG2, MFLG1)
85 C
86 C TOTAL TANK WEIGHT (O2 OR H2) (MAIN OR ACCUMULATORS)
87 WTOFTK(I) = WTG1 + WTG2
88 190 CONTINUE
89 RETURN
90 C
91 C CALCULATIONS FOR GENERAL TANK SHAPE INPUT
92 C MAIN STORAGE TANKS ONLY (IWOP = 2 OR 3)
93 C
94 200 KF = 1
95 ISV = NOSHAP + 1
96 C FIND FLUID TYPE OF BOTTOM TANK
97 KFL = IABS(JFLTP)
98 K1 = 1
99 K2 = 2
100 K3 = 1
101 IF (KFL .EQ. 1) GO TO 210
102 K1 = 2
103 K2 = 1
104 K3 = -1
105 C CALC. VOLUME AND AREA OF TANKS
106 210 DO 240 I=1, NOSHAP
107 KFL2 = IABS(JFLTP(I))
108 220 IF (KFL .EQ. KFL2) GO TO 230
109 C CHANGE IN FLUID TYPE
110 ISV = I
111 KF = 2
112 KFL = KFL2
113 C VOLUME FOR A PARTICULAR TANK SHAPE
114 230 TVL(I) = VFUNC (I)
115 TOTVL(KFL) = TOTVL(KFL) + TVL(I)

```

TNKHTA *****

```

116 C AREA FOR A PARTICULAR TANK SHAPE
117   TAR(I) = AFUNC (I)
118   240 CONTINUE
119 C HOW MANY SETS OF TANKS (1 OR 2)
120   IF (KF .EQ. 2) GO TO 250
121   K1 = KFL
122   K2 = K1
123   K3 = 1
124   250 IF (IWOP .NE. 3) GO TO 300
125 C
126 C FIT CYLINDRICAL SECTION (IWOP = 3)
127 C
128   IST = 1
129   DO 280 K=K1,K2,K3
130 C K IS THE FLUID TYPE FLAG
131   DO 260 I=IST,NOSHAP
132 C SEARCH LIST FOR CYLINDER (ONE MUST BE PRESENT AND
133 C THE HEIGHT SHOULD BE ZERO FOR IWOP = 3)
134   JCYL = I
135   IF (IABS(JTKTYP(I)) .EQ. 1) GO TO 270
136   260 CONTINUE
137 C CYLINDER IS AT JCYL IN LIST
138   270 TVL(JCYL) = TNKVL(K) - TOTVL(K)
139 C FIT CYLINDRICAL SECTION TO REQUIRED VOLUME
140   XD(JCYL) = TVL(JCYL) / (PI*YD(JCYL)**2)
141 C RE-CALCULATE AREA FOR FITTED CYLINDER
142   TAR(JCYL) = AFUNC (JCYL)
143   TOTVL(K) = TNKVL(K)
144   IST = JCYL + 1
145   280 CONTINUE
146   GO TO 320
147 C RE-CALCULATE PERCENT ULLAGE VOLUME (IWOP = 2 ONLY)
148   300 DO 310 K=K1,K2,K3
149 C TOTVL CALCULATED FROM THE INPUT DIMENSIONS
150   PCULLG(K) = ((TOTVL(K) - FLDVOL(K)) / TOTVL(K))*100.
151   310 TNKVL(K) = TOTVL(K)
152 C
153 C NOW CALCULATE HEAD FOR EACH TANK SET
154   320 IST = 1
155   IND = ISV - 1
156   DO 370 K=K1,K2,K3
157   THD(K) = 0.
158   TOTARA(K) = 0.
159   PV = 0.
160   HTGI = 0.
161 C
162   DO 340 I=IST,IND
163   PV = PV + TVL(I)
164   IF (PV .LT. FLDVOL(K)) GO TO 330
165   PVOL = FLDVOL(K) - PV + TVL(I)
166 C CALCULATE HEAD FOR THIS TANK SHAPE
167   THD(K) = THD(K) + HFUNC(I,PVOL)
168   GO TO 350
169   330 THD(K) = THD(K) + XD(I)
170   340 CONTINUE
171 C NOW CALCULATE TANK WEIGHTS
172   350 DO 360 I=IST,IND
173   JTKTP = IABS(JTKTYP(I))

```

```

*****  TNKHTA  *****
174      C
175      CALL THKWTG (2,K,JTKTP,MTYPE(K),TKTEMP(K),TKPRES(K),THD(K),TAR(I),
176      1      XD(I),YD(I),ZD(I),WTG2,MFLG1)
177      C
178      WTG1 = WTG1 + WTG2
179      C      CALC. TOTAL AREA THIS (O2 OR H2) TANK
180      TOTARA(K) = TOTARA(K) + TAR(I)
181      360 CONTINUE
182      C      TOTAL WEIGHT FOR THIS (O2 OR H2) TANK
183      WTOFTK(K) = WTG1
184      C      RESET INDEX FOR UPPER TANK
185      IST = ISV
186      IND = NOSHAP
187      370 CONTINUE
188      RETURN
189      END

```

SUBROUTINE TRAC

```

1      C
2      SUBROUTINE TRAC (NAME)
3      C
4      100 FORMAT (1H0 '** TRACE **' , A6)
5      C
6      WRITE (6,100) NAME
7      RETURN
8      END

```

***** FUNCTION TSAT

```

1      FUNCTION TSAT(P,I)
2      GO TO (1,2,3,2,3,6,7,6,7,10,3,12,13,14,15,16),I
3      1  T = 1519./((12.04-ALOG(P)))
4      GO TO 50
5      2  T = 225.74/(8.7137-ALOG(P)) - .95
6      IF(P.GT.19.)T=252.11/(9.4288-ALOG(P)) - .4
7      IF(T.GT.59.8) T = 59.8
8      GO TO 50
9      3  T = 1374./((11.63-ALOG(P)))
10     IF(P.GT.200.)T = 1763./((13.43-ALOG(P)))
11     GO TO 50
12     6  T = 1839./((11.83-ALOG(P)))
13     GO TO 50
14     7  T = 1050./((5.73-ALOG10(P)))
15     GO TO 50
16     10 T = 3168.7/(12.3579-ALOG(P))
17     GO TO 50
18     12 T = 5090./((14.45-ALOG(P)))
19     GO TO 50
20     13 T = 7348.3/(16.54098-ALOG(P))
21     GO TO 50
22     14 T = 6650./((13.4055-ALOG(P)))
23     GO TO 50
24     15 T = 1.8*1197./((7.4837-ALOG10(P/.01934)))
25     GO TO 50
26     16 T = 1.8*1996./((8.2875-ALOG10(P/.01934)))
27     50 TSAT=T
28     RETURN
29     END

```

***** FUNCTION TSAT

DATE

```

1      FUNCTION TSAT(P,I)
2      GO TO (1,2,3,2,3,6,7,6,7,10,3,12,13,14,15,16),I
3      T = 1519./((12.04-ALOG(P))
4      GO TO 50
5      2 T = 225.74/(8.7137-ALOG(P)) - .95
6      IF(P.GT.19.)T=252.11/(9.4288-ALOG(P)) - .4
7      IF(T.GT.59.8) T = 59.8
8      GO TO 50
9      3 T = 1374./((11.63-ALOG(P))
10     IF(P.GT.200.)T = 1763./((13.43-ALOG(P))
11     GO TO 50
12     6 T = 1839./((11.83-ALOG(P))
13     GO TO 50
14     7 T = 1050./((5.73-ALOG10(P))
15     GO TO 50
16     10 T = 3168.7/((12.3579-ALOG(P))
17     GO TO 50
18     12 T = 5090./((14.45-ALOG(P))
19     GO TO 50
20     13 T = 7348.3/((16.54098-ALOG(P))
21     GO TO 50
22     14 T = 6650./((13.4055-ALOG(P))
23     GO TO 50
24     15 T = 1.8*1197./((7.4837-ALOG10(P/.01934))
25     GO TO 50
26     16 T = 1.8*1996./((8.2875-ALOG10(P/.01934))
27     50 TSAT=T
28     RETURN
29     END

```


***** FUNCTION TSATH

1	FUNCTION TSATH(TEMP,HG,HL)	1783 C
2	DIMENSION R(19),TL(19),TG(19),TF(19)	1784 0
3	DATA R/1.022,2.0,4.0,8.0,14.0,25.0,43.0,69.0,99.0,128.0,151.0,	1785 0
4	1165.,176.0,182.0,185.0,186.5,187.25,187.46875,187.506/	1786 0
5	DATA	1787 0
6	227.07,29.81,33.07,36.18,39.96,44.12,48.33,51.97,54.79,56.72,57.80,	1788 0
7	358.57,58.99,59.18,59.29,59.34,59.353,59.356/	1789 0
8	DATA	1790 0
9	4,76.35,80.98,85.11,87.40,86.54,81.94,74.15,64.83,56.86,47.34,39.56	1791 0
10	5,33.46,28.34,22.31,18.66,16.55/	1792 0
11	DATA	1793 0
12	6,-110.86,-101.3,-89.04,-74.22,-58.58,-43.43,-30.07,-20.56,-11.13,	1794 0
13	7-4.27,1.17,5.54,10.83,14.29,16.36/	1795 0
14	T=TEMP	1796 0
15	IF(T.LT.24.845)T=24.845	1797 0
16	IF(T.GE.59.356)T=59.356	1798 0
17	DO 104 I=2,19	1799 0
18	IF(T-TF(I))102,101,104	1800 0
19	101 HL=TL(I)	1801 0
20	HG=TG(I)	1802 0
21	TSATH=R(I)	1803 0
22	RETURN	1804 0
23	102 D=TF(I)-TF(I-1)	1805 0
24	TRR=TF(I)-T	1806 0
25	TTR=T-TF(I-1)	1807 0
26	HL=(TL(I)*TTR+TL(I-1)*TRR)/D	1808 0
27	HG=(TG(I)*TTR+TG(I-1)*TRR)/D	1809 0
28	TSATH=(R(I)*TTR+R(I-1)*TRR)/D	1810 0
29	RETURN	1811 0
30	104 CONTINUE	1812 0
31	RETURN	1813 0
32	END	1814 0

***** SUBROUTINE TSIZEI

```

1      C      * * * * *
2      C      * ROUTINE NAME = INITIAL TANK SIZE ROUTINE *
3      C      * ROUTINE LANG = FORTRAN V UNIVAC 1108 EXEC 2*
4      C      * PROGRAMMER   = R. BOLLINGER 1943 102 26933 *
5      C      * DATE CODED   = 4/28/70
6      C      * REVISED     = JULY 1972
7      C      * PROGRAMMER   = J. MCKAY 01943 201 45178 *
8      C      * * * * *
9      C
10     C      SUBROUTINE TSIZEI(IFLG)
11     C
12     C      LOGICAL DIAG
13     C
14     C      INCLUDE CACCUM
15     C      INCLUDE CCNFIG
16     C      INCLUDE CENG
17     C      INCLUDE CHEX
18     C      INCLUDE CIOUNT
19     C      INCLUDE CMATRL
20     C      INCLUDE CMOTOR
21     C      INCLUDE CTANK
22     C      INCLUDE CTURBN
23     C
24     C      DIMENSION WGGHEX(2),WGGTP (2)
25     C
26     C      IF (DIAG(0,6HTSIZEI)) WRITE (107,6000) FLDLOD,SITEMP,SPGTEM,
27     C      1      SOPRES,SHFLUX,WGGTOT,(HXMRAT(1,1),I=1,2),
28     C      2      GWEGHT,THRATO,TIPWT,MIXRAT
29     C
30     C      ***** COMPUTE THE TOTAL PROPELLANT USE BY THE ENGINES, HEX
31     C      ***** AND GAS GENERATOR FOR EACH GAS (O2 AND H2)
32     C
33     C      H21 = WGGTOT(1)/(1.0 + HXMRAT(1,1))
34     C      H22 = WGGTOT(2)/(1.0 + HXMRAT(1,2))
35     C      WGGHEX(2) = H21 + H22
36     C      WGGHEX(1) = WGGTOT(1) - H21 + WGGTOT(2) - H22
37     C
38     C      H21 = GWEGHT(1)/(1.0 + THRATO(1))
39     C      H22 = GWEGHT(2)/(1.0 + THRATO(2))
40     C      WGGTP(2) = H21 + H22
41     C      WGGTP(1) = GWEGHT(1) - H21 + GWEGHT(2) - H22
42     C      IF(IFLG,EQ.2) GO TO 70
43     C
44     C      WETOT2 = TIPWT/(1.0 + MIXRAT)
45     C      WTTOT(2) = WETOT2 + .03 * WETOT2
46     C
47     C      WETOT1 = TIPWT - WTTOT(2)
48     C      WTTOT(1) = WETOT1 + .03 * WETOT1
49     C
50     C      GO TO 80
51     C
52     C      70 WTTOT(2) = TIPWT/(1.0 + MIXRAT)
53     C      WTTOT(1) = TIPWT - WTTOT(2)
54     C
55     C      80 CONTINUE
56     C
57     C      DO 6 I=1,2

```

```

***** TSIZE1 *****
58      IF (IFLG .EQ. 2) GO TO 4
59      C      FIRST PASS
60      WPTOT(II) = WTTOT(II) + WGGTP(II) + WGGHEX(II)
61      WTOTP(II) = WPTOT(II)
62      GO TO 6
63      C      SECOND PASS
64      4 WPTOT(II) = WPTOT(II) + WGGTP(II) + WTHXPG(II) + WGGPPG(II,1)
65      1      + WGR(II,1) + WLR(II) + WGRACC(II)
66      6 CONTINUE
67      C
68      IF (DIAG(2,6HTSIZE1)) WRITE (IOT,6000) WPTOT,WTTOT,WGGTP,WGGHEX
69      C
70      C      ***** COMPUTE THE VOLUME AND HEAT RATE FOR EACH TANK FOR
71      C      ***** EACH GAS WITHIN THE INPUT TANK CONFIGURATION.
72      C
73      C
74      DO 30 II=1,2
75      CALL FDNSTY (II,SITEMP(II,1),SIPRES(II,1),R)
76      C      HAS AN INITIAL LOAD BEEN INPUT
77      IF (FLDLOD(II)) 20,20,10
78      C      INITIAL LOAD INPUT -- USE IT
79      10 IF (IFLG .EQ. 2) GO TO 20
80      WTLFLUD = FLDLOD(II)
81      GO TO 30
82      C      PROPELLANT LOAD CALCULATED
83      20 WTLFLUD = WPTOT(II)
84      C      CALCULATE VOLUME OF FLUID
85      30 SVLFLD(II) = WTLFLUD / (R*NOP(II,1))
86      C
87      C      CALCULATE TANK WEIGHT AND AREA
88      CALL TNKHTA (I,SVLFLD,SULGPC,SHDIAM,SPGTEM,SOPRES,SHTYPE,SVOL,
89      1      TWT,TSA,TCYHT)
90      C
91      DO 60 II=1,2
92      IF (IFLG .EQ. 2) GO TO 40
93      C      FIRST PASS
94      SHRATE(II,1) = TSA(II,1)*SHFLUX(II,1) / 3600.
95      GO TO 50
96      C      SECOND PASS
97      40 IDXRI = SITYPE(II,1)
98      TIWT(II,1) = NOP(II,1)* TSA(II,1)*RHOI(IDXRI)*SITHIK(II,1) / 12.
99      50 TWT(II,1) = NOP(II,1)*TIWT(II,1)
100     60 CONTINUE
101     C
102     C      OUTPUT THE TANK SIZING DATA
103     C
104     CALL OTPTSZ (IFLG)
105     C
106     IF (IFLG .EQ. 1) RETURN
107     DO 100 II=1,2
108     IDX = INOXTK(II)
109     WI(IDX) = TIWT(II,1)
110     100 WEIGHT(IDX) = TWT(II,1)
111     RETURN
112     C
113     6000 FORMAT ('+',14X,6E15.6/(15X6E15.6))
114     C

```

***** TSIZEI *****

116

END

***** FUNCTION TSTART

```
1      FUNCTION TSTART(N)
2      C
3      DIMENSION T(17)
4      C
5      DATA (T(I),I=1,17)/162.3 , 36.5 ,153.04, 36.5 ,154. ,201. ,
6      1      231.4 ,201. ,231.4 ,325. ,153.04,432. ,
7      2      530.1 ,618.2 ,466.8 ,652.1 , 7.668/
8      C
9      TSTART = T(N)
10     RETURN
11     END
```

SUBROUTINE TVP

```

1      SUBROUTINE TVP(P,T)
2      COMMON /CTEVP/GT(8) /CRPR/CR(3)
3      COMMON /SCRH/ X(40)
4      C
5      C.... ROUTINE TO SOLVE VAPOR PRESSURE EQUATION ITERATIVELY FOR
6      C.... TEMPERATURE BY NEWTON'S METHOD
7      C
8      TC=CR(3)
9      C
10     C.... USE TEMP EXPLICIT EQN FOR FIRST APPROX
11     C
12     P2=P*P
13     P3=P2*P
14     P4=P3*P
15     P5=P4*P
16     P6=P5*P
17     X(1) = ALOG(P)
18     X(2) = 1.0
19     X(3)=P
20     X(4)=P2
21     X(5)=P3
22     X(5)=P3
23     X(6)=P4
24     X(7)=P5
25     X(8)=P6
26     T = 0.0
27     DO 1 I=1,8
28     1 T=X(I)*GT(I)
29     T = 1.0/T
30     C
31     C.... T IS NOW FIRST EST OF T
32     C
33     ITRMAX=25
34     EPS = 1.0E-7
35     DO 2 ITER=1,ITRMAX
36     PP=VPN(T)
37     CALL DPDTVP(T,P,DPDT)
38     DELTA=(P-PP)/DPDT
39     T=T+DELTA
40     IF(ABS(DELTA/T).LT.EPS) RETURN
41     2 CONTINUE
42     WRITE(6,300)P,T,DELTA
43     300 FORMAT('*** TVP DID NOT CONVERGE'//,
44     .
45     .
46     .
47     RETURN
48     END

```

***** SUBROUTINE TVPB

```
1      SUBROUTINE TVPB(PB,TB)
2      P = PB * 6.8947572E+3 / 1.01325E+5
3      CALL TVP(P,T)
4      TB = T * 1.8
5      RETURN
6      END
```

SUBROUTINE VARNAM

```
1      C
2      SUBROUTINE VARNAM
3      C
4      INCLUDE :CACCUH
5      INCLUDE :CAPU
6      INCLUDE :CCNFIG
7      INCLUDE :CCNTRL
8      INCLUDE :CDCYCL
9      INCLUDE :CENG
10     INCLUDE :CHEX
11     INCLUDE :CSYSWT
12     INCLUDE :CHSORC
13     INCLUDE :CIOUNT
14     INCLUDE :CKEYS
15     INCLUDE :CMATRL
16     INCLUDE :CMOTOR
17     INCLUDE :CNAMES
18     INCLUDE :CONST
19     INCLUDE :CPAGE
20     INCLUDE :CPUMP
21     INCLUDE :CTAB
22     INCLUDE :CTABA
23     INCLUDE :CTANK
24     INCLUDE :CTURBN
25     INCLUDE :DUMMY
26     INCLUDE :SPUMP
27     INCLUDE :TANKWT
28     C
29     RETURN
30     C
31     END
```

SUBROUTINE VENT

DATE 04177

```

1      SUBROUTINE VENT(Q,MH,MPV,ML,T,PV,PI,V,IG,PPVF,RHOP)
2      C
3      LOGICAL JP,DIAG
4      C
5      REAL MH,MPV,ML,MU,MT,MHF,MPVF,MTOT
6      C
7      INCLUDE TABLOK
8      C
9      C      THIS SUBROUTINE COMPUTES TOTAL VENT MASS DURING COAST
10     C      AS WELL AS REVISED VALUES OF LIQUID AND ULLAGE MASSES
11     C      AND TEMPERATURE FOR A MIXED FLUID SYSTEM
12     C
13     C      *** DEFINITION OF SYMBOLS ***
14     C
15     C      Q      TOTAL AMOUNT OF HEAT ADDED TO TANK (BTU)
16     C      MH      MASS OF HELIUM IN ULLAGE (LBM)
17     C      MPV     MASS OF PROPELLANT VAPOR IN ULLAGE (LBM)
18     C      ML      MASS OF LIQUID IN TANK (LBM)
19     C      T      TEMPERATURE OF FLUID (LIQUID + VAPOR) IN TANK (R)
20     C      PV      VENT PRESSURE (PSIA)
21     C      PI      INITIAL PRESSURE (PSIA)
22     C      V      TOTAL TANK VOLUME (CU FT)
23     C      IG      PROPELLANT FLAG (1=OXYGEN, 2=HYDROGEN)
24     C
25     C      ***** DIAG SWITCH IS PLACED HERE *****
26     C
27     IF (DIAG(0,6HVENT )) WRITE (6,6000) Q,MH,MPV,ML,T,PV,PI,V,IG,PPVF
28     C
29     C      *** INITIALIZE PRESSURE AND VENT MASS INCREMENTS ***
30     C
31     UHI=0.
32     UHF=0.
33     DPM=1.
34     DVTEST=1.0E-10
35     DEP=PI-PV
36     DP=DEP/10.
37     IF (DP.GT.DPM) GO TO 95
38     DP=AMINI(DEP,DPM)
39     95 GO TO (100,110),IG
40     100 CMWR=7.996
41     GC=48.25
42     GO TO 120
43     110 CMWR=0.50365
44     GC=766.
45     C
46     C      *** ENTER LOOP ON PRESSURE INCREMENT ***
47     C
48     120 DO 70 I=1,100
49     LOOP=0
50     DVMAX=0.99*(MPV,MH)
51     DMVMAX=DMVMAX
52     DMVMIN=0.000000001
53     C      *** CHECK MH ***
54     IF (MH.LE.0.) GO TO 1
55     PH=PI-PPVF
56     C      CALC. RHO OF GAS

```

```

***** VENT *****
58 CALL RHOLIQ(T,IG,RL)
59 VU=V-ML/RL
60 MHF = RHOG*VU
61 DIF=ABS(MHF-MH)
62 IF(DIF.GT.0.001) MH=MHF
63 I P=PI-DP
64 MU=MPV+MH
65 PB=0.5*(P+PI)
66 MT=MU+ML
67 C
68 C *** COMPUTE ENTHALPY OF VENT VAPOR ***
69 C
70 PG=PPVF*PB/PI
71 IF(MH.LE.0.) GO TO 3
72 CALL FINTAB (NTBID(35))
73 XTAB(1)=T
74 XTAB(2)=PH
75 HHI=MIPE(2,XTAB)
76 PH=(PI-PPVF)*(PB/PI)
77 XTAB(2)=PH
78 HHV=MIPE(2,XTAB)
79 C CALC. RHO OF GAS
80 3 CALL GSDNST (IG,T,PG,RG)
81 CALL FINTAB (NTBID(36)+IG)
82 XTAB(1)=RG
83 XTAB(2)=PG
84 HGI=MIPE(2,XTAB)
85 HV=(MPV/MU)*HGI+(MH/MU)*HHV
86 C
87 C *** COMPUTE NEW INTERNAL ENERGY FOR ASSUMED VENT MASS INCREMENT ***
88 C
89 KTAB = 0
90 IF (IG.EQ. 1 .AND. RHOP .LT. 40.) KTAB = 2
91 CALL FINTAB (NTBID(37)+IG+KTAB)
92 XTAB(1)=(ML+MPV)/V
93 XTAB(2)=PPVF
94 UI=MIPE(2,XTAB)*(ML+MPV)
95 C CALC. RHO OF GAS
96 CALL GSZDNS (IG,T,PPVF,RHOG,ZPV)
97 CALL ZFIND(T,PH,I7,ZH)
98 DEN=1.+(MH/MPV)*CHWR*(ZH/ZPV)
99 PPVF=P/DEN
100 MPVF = RHOG*VU
101 DMV=(MH+MPV)*(1.-MPVF/MPV)
102 DMVB=0.5*(DMVMAX+DMVMIN)
103 IF(DMV.GE.DMVMAX) DMV=DMVB
104 IF(DMV.LE.DMVMIN) DMV=DMVB
105 UHI=0.
106 IF(MH.LE.0.) GO TO 5
107 PH=PI*(1.-1./DEN)
108 UHI=MH*HHI-0.185*PH*VU
109 5 UI=UI+UHI
110 10 UFASS=UI+Q-DMV*HV
111 LOOP=LOOP+1
112 C
113 C *** COMPUTE PARTIAL PRESSURE OF PROPELLANT VAPOR ***
114 C
115 MHF=MH*(1.-DMV/MU)

```

```

***** VENT *****
116 MPVF=MPV*(1.-DMV/MU)
117 C
118 C *** COMPUTE INTERNAL ENERGY FOR THE NEW SATURATION CONDITIONS ***
119 C
120 KTAB = 0
121 IF (IG.EQ.1.AND.RHOP.LT.40.) KTAB = 2
122 CALL FINTAB (NTBID(38)+IG+KTAB)
123 XTAB(1)=(ML+MPVF)/V
124 XTAB(2)=PPVF
125 UF=MIPE(2,XTAB)*(ML+MPVF)
126 TF=TSAT(PPVF,IG)
127 IF(MHF.LE.0.) GO TO 15
128 PHF=P-PPVF
129 RH=MHF/VU
130 CALL ZFIND(TF,PHF,17,ZH)
131 PHF=ZFINDR(17)*RH*ZH*TF/144.
132 CALL FINTAB (NTBID(39))
133 XTAB(1)=TF
134 XTAB(2)=PHF
135 MHF=MIPE(2,XTAB)
136 UHF=(MHF*PHF-0.185*(P-PPVF)*VU
137 IF(UHF.LT.0.) LOOP=20
138 15 UF=UF+UHF
139 C *** CHECKOUT WRITE STATEMENTS ARE PLACED HERE ***
140 C
141 C *** COMPARE UF AND UFASS ***
142 C
143 DIF=UFASS-UF
144 DIFF=ABS(DIF)
145 IF(DIFF.LE.1.) GO TO 60
146 C
147 C *** REVISE VENT MASS INCREMENT ***
148 C
149 DMVC=DMVMAX-DMV
150 IF(DMVC.GT.DVTEST) GO TO 20
151 DP=0.5*DP
152 DP=0.5*DMVMAX
153 GO TO 70
154 20 IF(DIF) 30,60,40
155 30 DMVMAX=DMV
156 IF(DMVMAX.GT.DVTEST) GO TO 50
157 WRITE(6,1000) I,P,UFASS,UF,DMV
158 1000 FORMAT(31HOS/R VENT ANOMALY AT ITERATION ,I2,I3H DATA FOLLOW:IX,
159 14G13.8)
160 40 DMVMIN=DMV
161 50 DM=DMV-DIF*((DMV-DMVOLD)/(DIF-DIFOLD))
162 DMVOLD=DMV
163 DIFOLD=DIF
164 DMV=DM
165 IF(DMV.LE.DMVMAX.AND.DMV.GT.DMVMIN) GO TO 51
166 53 DMV=0.5*(DMVMAX+DMVMIN)
167 51 IF(LOOP.LT.20) GO TO 55
168 C *** CHECKOUT WRITE STATEMENTS ARE PLACED HERE ***
169 GO TO 60
170 55 GO TO 10
171 C
172 C *** VENT MASS ITERATION HAS CONVERGED. RECOMPUTE MASSES AND
173 C CHECK FOR PRESSURE CONVERGENCE. ***

```

```

***** VENT *****
174 60 PI=P
175 T=TF
176 MH=MHF
177 CALL RHOLIQ(T,IG,RL)
178 CALL ZFIND(T,PPVF,IG,ZPV)
179 RG=144.*PPVF/(ZPV*GC*T)
180 RFP=(ML+MPVF)/V
181 QUAL=(RL/RFP-1.)/(RL/RG-1.)
182 IF(QUAL.GT.0.) GO TO 65
183 C *** CHECKOUT WRITE STATEMENTS ARE PLACED HERE ***
184 QUAL=0.01
185 GO TO 66
186 65 IF(QUAL.LT.1.) GO TO 66
187 C *** CHECKOUT WRITE STATEMENTS ARE PLACED HERE ***
188 QUAL=1.
189 66 MTOT=ML+MPVF
190 MPV=QUAL*MTOT
191 ML=MTOT-MPV
192 C
193 IF(DIAG(2,6HVENT-1)) WRITE (6,6010) MH,MPV,ML,T,UF,UFASS,PPVF,P
194 C
195 DIF=P-PV
196 ADIF=DIF-.001
197 IF(ADIF.LE.0.) GO TO 80
198 IF(DIF.GT.DP) GO TO 70
199 DP=DIF
200 C
201 C ***** DIAG SWITCH IS PLACED HERE *****
202 C
203 70 CONTINUE
204 C
205 C *** CHECKOUT WRITE STATEMENTS ARE PLACED HERE ***
206 C
207 80 CONTINUE
208 C
209 C ***** DIAG SWITCH IS PLACED HERE *****
210 C
211 JP = DIAG(1,6HVENT )
212 C
213 RETURN
214 C
215 6000 FORMAT(1+1,14X,T23,1HEAT-IN1,T37,1HE-ULL1,T50,1PVAP-IN-ULL1,T65,
216 1 1LIQ-IN-TANK1,T82,1TFLUID1/15X,5E15.8/T20,1VENT-PRES1,T35,
217 2 1INIT-PRES1,T52,1TOT-VOL1,T67,1IGAS1,T80,1PRESSURE1/15X,3E15.8,
218 3 6X,13,6X,E15.8)
219 6010 FORMAT(1+1,14X,T23,1-MH-1,T37,1-MPV-1,T52,1-ML-1,T67,1-TFLUID-1,
220 1 /15X,4E15.8/T23,1-UF-1,T37,1-UFASS-1,T52,1-PPV-1,T67,1--P--1/15X,
221 2 4E15.8)
222 6020 FORMAT(39HOS/R VENT QUALITY ANOMALY AT ITERATION ,I2/
223 1 T23,1QUAL1,T37,1UF1,T52,1ULF1,T67,1UFG1/15X,4G13.8)
224 6030 FORMAT(101,14X,T23,1LOOP1,T37,1UFASS1,T52,1UF1,T67,1MHF1/
225 1 15X,115,3G15.8/T21,1MPVF1,T37,1TF1,T52,1PPVF1,T67,1PHF1,
226 2 T80,1DMV1,/15X,5G15.8)
227 C
228 END

```

***** SUBROUTINE VGVS

```

1      C      * * * * *
2      C      * ROUTINE NAME = MACH NUMBER COMPUTATION *
3      C      * ROUTINE LANG = FORTRAN V UNIVAC 1108 EXEC 2 *
4      C      * PROGRAMMER   = R. BOLLINGER 1963 102 26933 *
5      C      * DATE CODED   = 3/17/70 *
6      C      * * * * *
7      C
8      C      SUBROUTINE VGVS(IDX,RHO,IGAS)
9      C
10     C      ***** EXPLANATION OF THE CALLING SEQUENCE
11     C      *
12     C      * IDX  = INDEX OF THE CONFIGURATION TABLE
13     C      *
14     C      * RHO  = DENSITY OF THE GAS
15     C      *
16     C      ***** IGAS = GAS NUMBER (SEE S.R. FINDR)
17     C
18     C      INCLUDE CCONFIG
19     C      INCLUDE CONST
20     C
21     C      DATA IBLNK,IAST1,IAST6/'      ,,'      ,,'*****'/
22     C
23     C      ***** COMPUTE THE VELOCITY OF THE GAS
24     C
25     C      MFLG(IDX) = IBLNK
26     C      VG = 576. * WDOTN(IDX)/(PI * DIAM(IDX)**2 * RHO)
27     C
28     C      ***** COMPUTE THE VELOCITY OF SOUND IN GAS.
29     C
30     C      CALL CSUBP(TEMP(IDX),PRES(IDX),IGAS,CPGAS)
31     C      CVGAS = CSUBV(TEMP(IDX),PRES(IDX),IGAS)
32     C
33     C      VS = SQRT(GRAVTV*CPGAS*FINDR(IGAS)*TEMP(IDX)/CVGAS)
34     C
35     C      ***** COMPUTE MACH NUMBER
36     C
37     C      MACH(IDX) = VG/VS
38     C
39     C      ***** CHECH MACH NUMBER. FLAG MACH GREATER THAN .3 WITH ONE
40     C      ***** ASTRIK. FLAG MACH NUMBER GREATER THAN 1.0 WITH 6 ASTRIK
41     C
42     C      IF(MACH(IDX) = 0.3) 40,40,10
43     C      10 IF(MACH(IDX) = 1.0) 20,30,30
44     C      20 MFLG(IDX) = IAST1
45     C      GO TO 40
46     C      30 MFLG(IDX) = IAST6
47     C      40 RETURN
48     C      END

```

***** FUNCTION VPN

```

1      FUNCTION VPN(T)
2      COMMON/CVPN/G(11) /CRPR/CR(3)
3      COMMON /SCRH/ X(40)
4      C
5      C.... CALCULATE THE VAPOR PRESSURE
6      C
7      TC=CR(3)
8      A=G(11)
9      T2=T*T
10     T3=T*T2
11     T4=T*T3
12     T5=T*T4
13     T6=T*T5
14     X(1) = 1.0/T
15     X(2) = 1.0
16     X(3)=T
17     X(4)=T2
18     X(5)=T3
19     X(6)=T4
20     X(7)=T5
21     X(8)=T6
22     X(9) = ALOG(T)
23     X(10)=(TC-T)**A
24     P = 0.0
25     DO 1 I=1,10
26     1 P=P+X(I)*G(I)
27     P = EXP(P)
28     VPN=P
29     RETURN
30     END

```

***** FUNCTION VPNB

```
1      FUNCTION VPNB(TB)
2      T = TB/1.8
3      P=VPN(T)
4      VPNB = P * 1.01325E+5/6.8947572E+3
5      RETURN
6      END
```

***** SUBROUTINE VPROP

```

1      SUBROUTINE VPROP(T,P,D,K,H,S,U,Z)
2      COMMON /RFPR/ RF(10)
3      C
4      C.... ROUTINE TO CALCULATE THE PROPERTIES OF THE VAPOR
5      C
6      C      K =1   INPUT IS T + D
7      C      K =2   INPUT IS T + P
8      C      K =3   INPUT IS T, P, + D
9      C
10     IF(K.EQ.1)CALL PFND(T,D,P)
11     IF(K.EQ.2)CALL DFND(T,P,D,Z1,0)
12     HOTO=RF(1)
13     SOTO=RF(2)
14     RFST=RF(3)
15     RFHT=RF(4)
16     R   =RF(5)
17     AK  =RF(6)
18     FID=FING1(T,D)
19     F10 = FING1(T,0.0)
20     F2D=FING2(T,D)
21     F20 = FING2(T,0.0)
22     S0=SOTO+CP51(T)-CP51(RFST)
23     H0=HOTO+CPHI(T)-CPHI(RFHT)
24     S   = S0 - (R * ALOG(D*R*T)-FID + F10) * AK
25     H=H0+T*(FID-F10)+F2D-F20+P/D-R*T)*AK
26     U=H-(P/D)*AK
27     Z = P/(D*R*T)
28     RETURN
29     END

```

SUBROUTINE VPROP

```
1      SUBROUTINE VPROP(TB,PB,DB,K,HB,SB,UB,ZB)
2      COMMON /RFPR/RF(10)
3      WT=RF(7)
4      T = TB/1.8
5      P = PB * 6.8947572E+3/1.01325E+5
6      D = DB * 453.59237E-3/(WT * 2.8316847E-2)
7      CALL VPROP(T,P,D,K,H,S,U,Z)
8      PB = P * 1.01325E+5/6.8947572E+3
9      DB = D * WT * 2.8316847E-2/453.59237E-3
10     HB = H * 453.59237/(1.0543503E+3 * WT)
11     UB = U * 453.59237/(1.0543503E+3 * WT)
12     SB = S * 453.59237/(1.0543503E+3 * 1.8 * WT)
13     ZB = Z
14     RETURN
15     END
```

***** SUBROUTINE VSND

```

1      SUBROUTINE VSND(T,P,D,K,W)
2      COMMON /RFPR/ RF(10)
3      C
4      C,... ROUTINE TO CALCULATE THE SONIC VELOCITY FOR FOLLOWING INPUT OF K
5      C
6      C      K =1   INPUT IS T + P   RETURNS SONIC VELOCITY, W + D
7      C      K =2   INPUT IS T + D   RETURNS SONIC VELOCITY, W
8      C      K =3   INPUT IS T       RETURNS W, D, + P FOR SATURATED VAPOR
9      C      K =4   INPUT IS T       RETURNS W, D, + P FOR SATURATED LIQUID
10     C
11     AK=RF(6)
12     AM=RF(7)
13     IF((K.GT.0),OR,(K.LT.5))GO TO 1
14     WRITE(6,300)K
15     300 FORMAT(' *** ERROR IN CALL VSND ***',/,
16             '      K MUST EQUAL 1,2,3, OR 4',/,
17             '      K = ',I10)
18     RETURN
19     1 IF(K.EQ.2)GO TO 3
20     IF(K.GT.2)GO TO 2
21     CALL DFND(T,P,D,Z1,0)
22     GO TO 3
23     2 P=VPN(T)
24     IF(K.EQ.3)CALL DFND(T,P,D,Z1,2)
25     IF(K.EQ.4)CALL DFND(T,P,D,Z1,1)
26     3 CALL CPVTD(T,D,CP,CV)
27     W = (CP/CV) * DPDD(T,D) * (AK * 1000.0/AM)
28     IF(W.LE.0.0) GO TO 4
29     W = SQRT(W)
30     RETURN
31     4 CONTINUE
32     W = 0.0
33     RETURN
34     END

```

***** SUBROUTINE VSND8

```
1      SUBROUTINE VSND8(TB,PB,DB,K,WB)
2      COMMON /RFPR/ RF(10)
3      WT=RF(7)
4      T  = TB/1.8
5      P  = PB * 6.8947572E+3/.01325E+5
6      D  = DB * 453.59237E-3/(WT * 2.8316847E-2)
7      CALL VSND(T,P,D,K,W)
8      WB = W/3.048E-1
9      RETURN
10     END
```

***** SUBROUTINE WFIND

```

1      SUBROUTINE WFIND(T,W)
2      DIMENSION A(20)
3      DATA A/0.25084621E+3, 0.13509987E+3, 0.52785676E+2, 0.27676855E+2,
4      1      0.39105321E+2, 0.65561323E+2, 0.80803587E+2, 0.70524212E+2,
5      2      0.44784759E+2, 0.21252565E+2, 0.76797636E+1, 0.21368945E+1,
6      3      0.45984335E+0, 0.76361463E-1, 0.96932862E-2, 0.92306915E-3,
7      4      0.63811659E-4, 0.30229323E-5, 0.87755139E-7, 0.11770261E-8/
8
9      C.... ROUTINE TO CALCULATE W(IPTS-68(T68)) GIVEN TEMP
10     C
11     C.... CALCULATION DONE BY NEWTON'S METHOD
12     C
13     ITERMX=100
14     EPS = 1.0E-4
15
16     C.... CALC FIRST ESTIMATE
17     C
18     W = 0.40713654E-2 * T = 0.11209346E+0
19     DO 1 ITER=1,ITERMX
20     DLW = ALOG(W)
21
22     C.... CALC T GIVEN EST OF W
23     C
24     TT = 0.0
25     DO 2 I=1,20
26     2 TT=TT+A(I)*DLW**I
27     TT = TT + 273.15
28
29     C.... CALC DT/DW
30     C
31     DTDW = 0.0
32     DO 3 I=1,20
33     AI=1
34     3 DTDW=DTDW+AI*A(I)*DLW**(I-1)/W
35
36     C
37     DELTA=(T-TT)/DTDW
38     W=W+DELTA
39     IF (ABS(DELTA).LT.EPS) RETURN
40     1 CONTINUE
41     WRITE(6,100)DELTA
42     100 FORMAT(' *** WFIND FAILED TO CONVERGE - DELTA =',G20.10)
43     RETURN
44     END

```

***** SUBROUTINE WTACC

```

1      C      * * * * *
2      C      * ROUTINE NAME - ACCUMULATOR WEIGHTS *
3      C      * DETERMINATION ROUTINE *
4      C      * ROUTINE LANG - FORTRAN V UNIVAC 1108 EXEC 2 *
5      C      * PROGRAMMER   - R. BOLLINGER 1943 102 26933 *
6      C      * DATE CODED   - 5/20/70 *
7      C      * REVISED     - JULY 1972 *
8      C      * PROGRAMMER   - J. MCKAY D1943 201 45178 *
9      C      * * * * *
10     C
11     C      SUBROUTINE WTACC
12     C
13     C      INCLUDE CACCUM
14     C      INCLUDE CCONFIG
15     C      INCLUDE CMATRL
16     C
17     C      DIMENSION ADUMMY(2),ZERO(2)
18     C
19     C      DATA ZERO / 0.,0./
20     C
21     C      ***** COMPUTE ACCUMULATOR WEIGHT AND SURFACE AREA.
22     C
23     C      CALL TNKHTA (2,AVOL,ZERO,ADIAM,ATEMP,APRES,AMTYPE,ADUMMY,ACHT,AA,
24     C      I ACYHT)
25     C      DO 10 I=1,2
26     C      IAI = AITYPE(I)
27     C      CALC. ACCUMULATOR INSULATION WEIGHT
28     C      ACIWT(I) = NAOP(I)*AA(I)*RHOI(IAI)*AITHIK(I) / 12.
29     C      ACHT(I) = NAOP(I)*ACHT(I)
30     C      IDX = INDXAC(I)
31     C      WEIGHT(IDX) = ACHT(I)
32     C      WI(IDX) = ACIWT(I)
33     C      10 CONTINUE
34     C      PRINT ACCUMULATOR DATA
35     C      CALL OTPACC
36     C
37     C      RETURN
38     C      END

```

***** FUNCTION YLGINT

```

1      FUNCTION YLGINT(X,Y,N,XBAR,NPTS,S)          YLGINT
2      C-----YLGINT
3      C   LAGRANGE INTERPOLATION                YLGINT
4      C-----YLGINT
5      C   ARGUMENT DEFINITION                  YLGINT
6      C   X   -- ARRAY OF N INDEPENDENT VARIABLE VALUES(INPUT)  YLGINT
7      C   Y   -- ARRAY OF N DEPENDENT VARIABLE VALUES (INPUT)  YLGINT
8      C   N   -- NUMBER OF DATA POINTS (INPUT)                 YLGINT
9      C   XBAR -- INDEPENDENT VARIABLE VALUE FOR WHICH YLGINT IS THE YLGINT
10     C   INTERPOLANT (INPUT)                                YLGINT
11     C   NPTS -- NUMBER OF DATA POINTS TO BE USED IN INTERPOLATION(INPUT) YLGINT
12     C   S   -- ERROR EXIT IF N IS LESS THAN 2, OR IF OVERFLOW IS YLGINT
13     C   DETECTED.                                          YLGINT
14     C-----YLGINT
15     C   REFERENCE                                         YLGINT
16     C   K. S. KUNZ, NUMERICAL ANALYSIS, MCGRAW-HILL BOOK COMPANY, 1957. YLGINT
17     C-----YLGINT
18     C   DIMENSION X(N),Y(N)                             YLGINT
19     C-----YLGINT
20     C   CHECK ARGUMENTS N AND NPTS                    YLGINT
21     C   IF NPTS IS OUT OF RANGE, USE ALL POINTS        YLGINT
22     C-----YLGINT
23     C   IF(N.LT.2) RETURN 6                             YLGINT
24     C   IF(NPTS.GT.1.AND.NPTS.LT.N) GO TO 2             YLGINT
25     C   NPTS=N                                           YLGINT
26     C   1 JL=1                                           YLGINT
27     C   JH=NPTS                                          YLGINT
28     C   GO TO 9                                           YLGINT
29     C-----YLGINT
30     C   DETERMINE POSITION OF XBAR WITHIN X ARRAY        YLGINT
31     C   UPON EXIT FROM LOOP 3, X(JH-1) LT XBAR LE X(JH) YLGINT
32     C-----YLGINT
33     C   2 DO 3 JH=1,N                                    YLGINT
34     C   IF(XBAR-X(JH)) 5,12,3                             YLGINT
35     C   3 CONTINUE                                       YLGINT
36     C   4 JH=N                                           YLGINT
37     C   JL=N-NPTS+1                                      YLGINT
38     C   GO TO 9                                           YLGINT
39     C-----YLGINT
40     C   DETERMINE WHICH POINTS USED IN INTERPOLATION   YLGINT
41     C   THE POINTS CHOSEN ARE THE ONES WHOSE ABSCISSAS ARE CLOSEST TO XBAR YLGINT
42     C   JL IS LOWEST SUBSCRIPT OF POINTS USED, JH IS HIGHEST SUBSCRIPT YLGINT
43     C-----YLGINT
44     C   5 JL=JH                                           YLGINT
45     C   D2=X(JH)-XBAR                                     YLGINT
46     C   6 JL=JL-1                                         YLGINT
47     C   IF(JL.LE.1) GO TO 1                               YLGINT
48     C   IF(NPTS.EQ.2) GO TO 99                           YLGINT
49     C   D1=XBAR-X(JL)                                     YLGINT
50     C   7 IF(JH-JL+1.EQ.NPTS) GO TO 8                    YLGINT
51     C   IF(XBAR-X(JL-1).LT.D2) GO TO 6                   YLGINT
52     C   JH=JH+1                                           YLGINT
53     C   IF(JH.GE.N) GO TO 4                               YLGINT
54     C   D2=X(JH)-XBAR                                     YLGINT
55     C   GO TO 7                                           YLGINT
56     C   8 INDCT=0                                         YLGINT
57     C   IF(JL.GT.1.AND.XBAR-X(JL-1).LT.D2) INDCT=-1

```

B-332

***** FUNCTION YLNTRP

```
1      C
2      C      ROUTINE TO INTERPOLATE ON LOG-LOG
3      C      COORDINATES, FOR TABLES WHICH HAVE
4      C      BEEN INPUT LN(X),LNCY) FORM (6-8-71)
5      C
6      C      FUNCTION YLNTRP (NV,XVAL)
7      C      NOT USED -- DELETE
8      YLNTRP = 1.0E+10
9      RETURN
10     END
```

SUBROUTINE ZFIND

```

1      SUBROUTINE ZFIND(T,P,N,V)
2      DIMENSION G(3,17),S(17)
3      DIMENSION A(17,6),TS(17)
4      DATA (TS(K),K=1,16)
5      1 /150., 30., 140., 30., 140., 190., 220., 190.,
6      2 220., 300., 140., 370., 475., 480., 450., 450./
7      DATA(G(1,1),I=1,17)/277.85,59.8,259.13,59.8,260.,343.2,387.,343.2
8      1,387.,521.8,259.13,730.,776.4,1094.,749.5,1155.,9.37/
9      DATA(G(2,1),I=1,17)/743.78,187.7,822.8,187.7,795.,673.1,719.,673.
10     1 1.719.,581.,822.8,1652.,1470.,1696.,771.,1470.,33.82/
11     DATA(G(3,1),I=1,17)/48.31,766.8,40.67,766.8,42.01,96.35,28.62,
12     1 96.35,28.62,55.81,40.67,90.77,16.78,37.0,11.90,33.50,386.3/
13     DATA S/5HL02 .5HLH2 .5HLF2 .5HLH2 .5HFL0X .5HCH4 .5HOF2 .
14     1 .5HCH4 .5HOF2 .5HAR2H6 .5HLF2 .5HNM3 .5HN204 .5HA=50 .
15     2 .5HCLF=5.5HMHF=5.5HHE /
16     DATA(A( 1,J),J=1,6)/.2142592E1,.,3228322E-1,.,3563987E-3,
17     1 -.1895669E-5,4823166E-8,.,5002791E-11/
18     DATA(A( 2,J),J=1,6)/-.4458459E1,.,6350202,.,2863016E-1,
19     1 .6260532E-3,.,6704223E-5,.,2763409E-7/
20     DATA(A( 3,J),J=1,6)/.19209203E1,.,4596045E-1,.,7505833E-3,
21     1 -.5522453E-5,1900218E-7,.,2526448E-10/
22     DATA(A( 4,J),J=1,6)/-.4458459E1,.,6350202,.,2863016E-1,
23     1 .6260532E-3,.,6704223E-5,.,2763409E-7/
24     DATA(A( 5,J),J=1,6)/.18476612E1,.,4146008E-1,.,6702877E-3,
25     1 -.4895455E-5,1674002E-7,.,2217407E-10/
26     DATA(A( 6,J),J=1,6)/.17005803E1,.,1669025E-1,.,1536145E-3,
27     1 -.6691577E-6,13609972E-8,.,11357811E-11/
28     DATA(A( 7,J),J=1,6)/.15219822E2,.,28012466,.,22001021E-2,
29     1 -.86119226E-5,16835339E-7,.,1326415E-10 /
30     DATA(A( 8,J),J=1,6)/.17005803E1,.,1669025E-1,.,1536145E-3,
31     1 -.6691577E-6,13609972E-8,.,11357811E-11/
32     DATA(A( 9,J),J=1,6)/.15219822E2,.,28012466,.,22001021E-2,
33     1 -.86119226E-5,16835339E-7,.,1326415E-10 /
34     DATA(A(10,J),J=1,6)/-.81449607E-1,.,15604836E-1,.,91954274E-4,
35     1 .27813987E-6,.,42739698E-9,.,25170512E-12/
36     DATA(A(11,J),J=1,6)/.19209203E1,.,4596045E-1,.,7505833E-3,
37     1 -.5522453E-5,1900218E-7,.,2526448E-10/
38     DATA(A(12,J),J=1,6)/.39233318E1,.,34565291E-1,.,16306507E-3,
39     1 -.39452432E-6,45572795E-9,.,21979859E-12/
40     DATA(A(13,J),J=1,6)/.70122306E1,.,57413097E-1,.,21967497E-3,
41     1 -.42198721E-6,40864988E-9,.,1606916E-12/
42     DATA(A(14,J),J=1,6)/.86405843E1,.,72176161E-1,.,2724231E-3,
43     1 -.5137489E-6,4844758E-9,.,18308062E-12/
44     DATA(A(15,J),J=1,6)/.54858839E1,.,44806287E-1,.,17789492E-3,
45     1 -.35089558E-6,34539726E-9,.,13832016E-12/
46     DATA(A(16,J),J=1,6)/.49407545E1,.,39649956E-1,.,15955648E-3,
47     1 -.32115667E-6,32356706E-9,.,13068156E-12/
48
49     C IF(P.GT.0.0.AND.T.GT.0.0) GO TO 666
50     WRITE (6,777) P,T
51     777 FORMAT(5X,'ENTERING ZFIND A PRESSURE OR A TEMPERATURE IS OUT OF RA
52     INGE'/TS,'PRESSURE '=I,E15.8,T30,'TEMPERATURE '=I,E15.8)
53     CALL EXIT
54     666 CONTINUE
55
56     C IF(N.EQ.17) GO TO 56
57     IF(T.GT.650..OR.T.LT.25.) GO TO 35

```

```

***** ZFIND *****
58      IF(T.GT.TS(N)+100.,OR.T.LT.TS(N)) GO TO 56
59      C**** TEST TO SEE IF SAT,D COMP,Y Z APPLIES *****
60      CALL PVAPOR(T,N,TRYP)
61      TRY=TRYP-P
62      IF(ABS(TRY).LT.5.) GO TO 55
63      56 CONTINUE
64      C*****REDLICH-KWONG*****AGO-GO*****0064
65      VF=G(3,N)*T/(P*144.)
66      IF(N.EQ.2,OR.N.EQ.4) GO TO 10
67      AS=.4278*G(3,N)*G(3,N)/(G(2,N)*144.)*G(1,N)**2.5
68      BS=.0867*G(3,N)*G(1,N)/(G(2,N)*144.)
69      IN=0
70      N2=0
71      E=.00001
72      V=VF
73      199 Y=G(3,N)*T/(V-BS)-AS/(T*.5*V*(V+BS))-P*144.
74      IF(ABS(Y).LT.E) GO TO 200
75      C*****FIRST DERIVITIVE OF REDLICH-KWONG RESPECT TO V*****0075
76      YP=-G(3,N)
77      1 :T/((V-BS)*(V-BS))+AS*(2.*V+BS)/(T*.5*V*V*(V+BS)*(V+BS))
78      IN=IN+1
79      N2=N2+1
80      IF(N2.EQ.25) E=2.*E
81      IF(N2.EQ.25) N2=0
82      IF(IN.GT.1000) GO TO 201
83      V=V-YP
84      IF(V.GT.0.) GO TO 43
85      OLDV=V+Y/YP
86      DELTA=Y/YP
87      FACTOR=.05
88      44 V=OLDV-FACTOR*DELTA
89      FACTOR=FACTOR*.9
90      IF(V.LT.0.) GO TO 44
91      43 CONTINUE
92      GO TO 199
93      200 CONTINUE
94      GO TO 50
95      201 CONTINUE
96      WRITE(6,40) T,P,S(N),V,VF
97      40 FORMAT(1X,34HREDLICH = KWONG      FLUNKED T = ,F4.2,X,4HP = ,
98      1 F7.2.
99      2 5H FOR ,A5,2X,4HV = ,E10.5,2X,8HRETURNED,E10.5)
100     V=VF
101     GO TO 50
102     10 TT=T+0.5
103     V=PTDENS(P ,TT)
104     V=1./V
105     50 V=V/VF
106     RETURN
107     15 V=1.
108     RETURN
109     55 CONTINUE
110     V=A(N,1)+A(N,2)*T+A(N,3)*T*T+A(N,4)*T*T*T+A(N,5)*T**4+A(N,6)*T**5
111     RETURN
112     END

```

***** FUNCTION ZGET

```
1      FUNCTION ZGET(T,P,IGAS)
2
3      C
4      C      ROUTINE CALLS STEWARTS COMBINED O2-N2 PROPERTIES PROGRAM
5      C
6      CALL ONPROP(T,P,DX,ENX, SX,UX,ZX,IGAS)
7      ZGET = ZX
8      RETURN
9      END
```

Appendix B
THE CRYOGENIC INTEGRATED MATH MODEL
(TCIMM)

PART II - PROGRAM FILE ELEMENT TABLE OF CONTENTS

The next several pages contain the program file table of contents, also known as the PRT,T index. The table is printed when called for by a ATPRT,T control card. The output contains the "element table," "procedure tables," and, if a ATPREP card has preceded the PRT card, the "entry point table."

The column headings given at the beginning of the element table have the following meanings:

D-FLAG	- an asterisk means that the entry is deleted from the file.
NAME	- name of symbolic/relocatable/absolute element.
VERSION	- version of element.
TYPE	- if the element is symbolic, the processor which created it is indicated.
DATE, TIME	- time that element was added to the file.
SEQUENCE NO.	- position of the element in the file. This is sequentially issued as elements are added to the file.
PRE-SIZE	- for relocatable elements, the preamble length is given in sectors (28 words per sector).
TEXT-SIZE	- this is the text size in sectors.
CYCLE WORD	- the cycle word is broken up into three separate parameters; starting from left to right, they are: <ul style="list-style-type: none"> (1) the number of cycles the system will maintain (2) the number of the most current cycles (absolute scale) (3) the number of cycles currently being maintained.
LOCATION	- refers to the sector position relating to the start of the file (1792 is the base).

It should be noted that the entry point table is fugitive in the sense that it must be recreated each time a change is made in the program, and is subject to the following constraints:

- (1) Destroyed when an update is made to any element in a program file.
- (2) Destroyed when program file is put on magnetic tape.
- (3) Is not re-established when file is copied from tape to drum.
- (4) Contains externalized labels.
- (5) Is created by the ATPREP statement which will prepare or re-establish an entry point table for a specified program file.

TABLE OF CONTENTS - FILE TCIMM.

D	NAME	VERSION	TYPE	DATE	TIME	SEQ #	SIZE-PR,TEXT	(CYCLE	WORD)	PSRMODE	LOCATION
	CACCU		FOR PROC	13 MAR 73	01:51:49	1	14	1	0		1792
	CAPU		FOR PROC	13 MAR 73	01:51:50	2	50	1	0		1806
	CCNFIG		FOR PROC	13 MAR 73	01:51:51	3	43	1	0		1856
	CCNTRL		FOR PROC	13 MAR 73	01:51:52	4	10	1	0		1899
	CCCYCL		FOR PROC	13 MAR 73	01:51:52	5	10	1	0		1909
	CENG		FOR PROC	13 MAR 73	01:51:54	6	10	1	0		1919
	CFLRAT		FOR PROC	13 MAR 73	01:51:54	7	3	1	0		1929
	CFLUID		FOR PROC	13 MAR 73	01:51:54	8	1	1	0		1932
	CHEX		FOR PROC	13 MAR 73	01:51:55	9	22	1	0		1933
	CHTX		FOR PROC	13 MAR 73	01:51:56	10	4	1	0		1955
	CHSORC		FOR PROC	13 MAR 73	01:51:57	11	13	1	0		1959
	COUNT		FOR PROC	13 MAR 73	01:51:57	12	5	1	0		1972
	CKEYS		FOR PROC	13 MAR 73	01:51:58	13	1	1	0		1977
	CMATRL		FOR PROC	13 MAR 73	01:51:58	14	17	1	0		1978
	CMOTOR		FOR PROC	13 MAR 73	01:51:59	15	2	1	0		1995
	CONST		FOR PROC	13 MAR 73	01:52:00	16	2	1	0		1997
	CPAGE		FOR PROC	13 MAR 73	01:52:00	17	8	1	0		1999
	CPUMP		FOR PROC	13 MAR 73	01:52:01	18	20	1	0		2007
	CSYSWT		FOR PROC	13 MAR 73	01:52:01	19	6	1	0		2027
	CTAR		FOR PROC	13 MAR 73	01:52:02	20	10	1	0		2033
	CTARA		FOR PROC	13 MAR 73	01:52:02	21	2	1	0		2043
	CTANK		FOR PROC	13 MAR 73	01:52:03	22	19	1	0		2045
	CTURBN		FOR PROC	13 MAR 73	01:52:03	23	13	1	0		2064
	DUMMY		FOR PROC	13 MAR 73	01:52:04	24	3	1	0		2077
	SPUMP		FOR PROC	13 MAR 73	01:52:04	25	2	1	0		2080
	TABLOK		FOR PROC	13 MAR 73	01:52:06	26	1	1	0		2082
	TANKWT		FOR PROC	13 MAR 73	01:52:06	27	3	1	0		2083
	ACCRES		RELOCATABLE	13 MAR 73	01:52:09	28	1	3			2086
	ACCRES		FOR SYMB	13 MAR 73	01:52:09	29	7	5	0		2090
	ACQWT		RELOCATABLE	13 MAR 73	01:52:13	30	2	3			2097
	ACQWT		FOR SYMB	13 MAR 73	01:52:13	31		10	5	0	2102
	ALPHAB		RELOCATABLE	13 MAR 73	01:52:15	32	1	1			2112
	ALPHAB		FOR SYMB	13 MAR 73	01:52:15	33		3	5	0	2114
	APUFLO		RELOCATABLE	13 MAR 73	01:52:20	34	3	15			2117
	APUFLO		FOR SYMB	13 MAR 73	01:52:20	35		31	5	0	2135
	APUSUP		RELOCATABLE	13 MAR 73	01:52:49	36	5	100			2166

APUSUP	FOR SYMB	13 MAR 73	01:52:49	37		156	5	0	1	2271
BETAB	RELOCATABLE	13 MAR 73	01:52:52	38	1	3				2427
BETAB	FOR SYMB	13 MAR 73	01:52:52	39		5	5	0	1	2431
CFTW	RELOCATABLE	13 MAR 73	01:52:55	40	1	9				2436
CFTW	FOR SYMB	13 MAR 73	01:52:55	41		7	5	0	1	2446
COMFLO	RELOCATABLE	13 MAR 73	01:53:18	42	2	15				2453
COMFLO	FOR SYMB	13 MAR 73	01:53:18	43		25	5	0	1	2470
CONSUM	RELOCATABLE	13 MAR 73	01:54:05	44	2	2				2495
CONSUM	FOR SYMB	13 MAR 73	01:54:06	45		3	5	0	1	2499
CPHI	RELOCATABLE	13 MAR 73	01:54:10	46	2	6				2502
CPHI	FOR SYMB	13 MAR 73	01:54:11	47		5	5	0	1	2510
CPIG	RELOCATABLE	13 MAR 73	01:54:13	48	2	5				2515
CPIG	FOR SYMB	13 MAR 73	01:54:13	49		6	5	0	1	2522
CPSI	RELOCATABLE	13 MAR 73	01:54:16	50	2	6				2528
CPSI	FOR SYMB	13 MAR 73	01:54:16	51		5	5	0	1	2536
CPVTDB	RELOCATABLE	13 MAR 73	01:54:18	52	1	3				2541
CPVTDB	FOR SYMB	13 MAR 73	01:54:18	53		3	5	0	1	2545
CPVTD	RELOCATABLE	13 MAR 73	01:54:21	54	2	8				2548
CPVTD	FOR SYMB	13 MAR 73	01:54:21	55		6	5	0	1	2558
CRYCON	RELOCATABLE	13 MAR 73	01:54:25	56	3	9				2564
CRYCON	FOR SYMB	13 MAR 73	01:54:25	57		15	5	0	1	2576
CSPF21	RELOCATABLE	13 MAR 73	01:54:27	58	1	5				2591
CSPF21	FOR SYMB	13 MAR 73	01:54:27	59		4	5	0	1	2597
CSURP	RELOCATABLE	13 MAR 73	01:54:30	60	2	8				2601
CSURP	FOR SYMB	13 MAR 73	01:54:30	61		11	5	0	1	2611
CSURPV	RELOCATABLE	13 MAR 73	01:54:32	62	2	4				2622
CSURPV	FOR SYMB	13 MAR 73	01:54:32	63		5	5	0	1	2628
CSURP1	RELOCATABLE	13 MAR 73	01:54:35	64	1	11				2633
CSURP1	FOR SYMB	13 MAR 73	01:54:36	65		17	5	0	1	2645
CSURV	RELOCATABLE	13 MAR 73	01:54:38	66	2	6				2662
CSURV	FOR SYMB	13 MAR 73	01:54:38	67		9	5	0	1	2670
DATAN2	RELOCATABLE	13 MAR 73	01:54:43	68	3	16				2679
DATAN2	FOR SYMB	13 MAR 73	01:54:43	69		36	5	0	1	2698
DATA02	RELOCATABLE	13 MAR 73	01:54:48	70	3	16				2734
DATA02	FOR SYMB	13 MAR 73	01:54:48	71		35	5	0	1	2753
DCALC	RELOCATABLE	13 MAR 73	01:55:01	72	1	15				2788
DCALC	FOR SYMB	13 MAR 73	01:55:02	73		16	5	0	1	2804
DENSON	RELOCATABLE	13 MAR 73	01:55:08	74	2	3				2820
DENSON	FOR SYMB	13 MAR 73	01:55:08	75		5	5	0	1	2825
DFNDB	RELOCATABLE	13 MAR 73	01:55:10	76	1	3				2830
DFNDB	FOR SYMB	13 MAR 73	01:55:10	77		3	5	0	1	2834
DFND	RELOCATABLE	13 MAR 73	01:55:12	78	2	11				2837
DFND	FOR SYMB	13 MAR 73	01:55:13	79		9	5	0	1	2850
DIAG	RELOCATABLE	13 MAR 73	01:55:15	80	1	5				2859
DIAG	FOR SYMB	13 MAR 73	01:55:15	81		11	5	0	1	2865
DPDDB	RELOCATABLE	13 MAR 73	01:55:16	82	1	3				2876
DPDDB	FOR SYMB	13 MAR 73	01:55:16	83		3	5	0	1	2880
DPDD	RELOCATABLE	13 MAR 73	01:55:21	84	2	15				2883
DPDD	FOR SYMB	13 MAR 73	01:55:21	85		12	5	0	1	2900
DPDT8	RELOCATABLE	13 MAR 73	01:55:22	86	1	2				2912
DPDT8	FOR SYMB	13 MAR 73	01:55:22	87		2	5	0	1	2915
DPDT	RELOCATABLE	13 MAR 73	01:55:26	88	2	12				2917
DPDT	FOR SYMB	13 MAR 73	01:55:27	89		11	5	0	1	2931
DPDTVP	RELOCATABLE	13 MAR 73	01:55:31	90	2	6				2942
DPDTVP	FOR SYMB	13 MAR 73	01:55:31	91		5	5	0	1	2950
DSATL	RELOCATABLE	13 MAR 73	01:55:39	92	2	4				2955
DSATL	FOR SYMB	13 MAR 73	01:55:39	93		5	5	0	1	2961
DSATV	RELOCATABLE	13 MAR 73	01:55:40	94	2	4				2966

DSATV	FOR SYMB	13 MAR 73	01:55:41	95		5	5	0	1	2972
ENGINE	RELOCATABLE	13 MAR 73	01:56:11	96	3	17				2977
ENGINE	FOR SYMB	13 MAR 73	01:56:11	97		22	5	0	1	2997
FINDR	RELOCATABLE	13 MAR 73	01:56:13	98	1	3				3019
FINDR	FOR SYMB	13 MAR 73	01:56:13	99		2	5	0	1	3023
FING1	RELOCATABLE	13 MAR 73	01:56:16	100	2	14				3025
FING1	FOR SYMB	13 MAR 73	01:56:16	101		12	5	0	1	3041
FING2	RELOCATABLE	13 MAR 73	01:56:20	102	2	14				3053
FING2	FOR SYMB	13 MAR 73	01:56:20	103		11	5	0	1	3069
FING3	RELOCATABLE	13 MAR 73	01:56:24	104	2	14				3080
FING3	FOR SYMB	13 MAR 73	01:56:24	105		12	5	0	1	3096
FINTAB	RELOCATABLE	13 MAR 73	01:56:27	106	2	9				3108
FINTAB	FOR SYMB	13 MAR 73	01:56:27	107		12	5	0	1	3119
FLORAT	RELOCATABLE	13 MAR 73	01:56:32	108	3	22				3131
FLORAT	FOR SYMB	13 MAR 73	01:56:33	109		28	5	0	1	3156
FLODEQ	RELOCATABLE	13 MAR 73	01:56:34	110	1	3				3184
FLODEQ	FOR SYMB	13 MAR 73	01:56:34	111		8	5	0	1	3188
GASGEN	RELOCATABLE	13 MAR 73	01:57:10	112	2	10				3196
GASGEN	FOR SYMB	13 MAR 73	01:57:10	113		15	5	0	1	3208
GETCON	RELOCATABLE	13 MAR 73	01:57:14	114	2	4				3223
GETCON	FOR SYMB	13 MAR 73	01:57:14	115		8	5	0	1	3229
GOMTRY	RELOCATABLE	13 MAR 73	01:57:18	116	3	28				3237
GOMTRY	FOR SYMB	13 MAR 73	01:57:18	117		14	5	0	1	3268
HEATEX	RELOCATABLE	13 MAR 73	01:57:30	118	3	67				3282
HEATEX	FOR SYMB	13 MAR 73	01:57:31	119		89	5	0	1	3352
HEXELC	RELOCATABLE	13 MAR 73	01:57:36	120	2	14				3441
HEXELC	FOR SYMB	13 MAR 73	01:57:36	121		24	5	0	1	3457
HTLEAK	RELOCATABLE	13 MAR 73	01:57:42	122	1	1				3481
HTLEAK	FOR SYMB	13 MAR 73	01:57:42	123		2	5	0	1	3483
HPTCP	RELOCATABLE	13 MAR 73	01:57:44	124	1	2				3485
HPTCP	FOR SYMB	13 MAR 73	01:57:45	125		3	5	0	1	3488
HPTCV	RELOCATABLE	13 MAR 73	01:57:46	126	1	2				3491
HPTCV	FOR SYMB	13 MAR 73	01:57:46	127		3	5	0	1	3494
HPTGAM	RELOCATABLE	13 MAR 73	01:57:48	128	1	2				3497
HPTGAM	FOR SYMB	13 MAR 73	01:57:48	129		3	5	0	1	3500
HPW	RELOCATABLE	13 MAR 73	01:57:50	130	1	1				3503
HPW	FOR SYMB	13 MAR 73	01:57:50	131		4	5	0	1	3505
HVAP	RELOCATABLE	13 MAR 73	01:57:55	132	1	8				3509
HVAP	FOR SYMB	13 MAR 73	01:57:55	133		5	5	0	1	3518
HYENTH	RELOCATABLE	13 MAR 73	01:58:10	134	1	89				3523
HYENTH	FOR SYMB	13 MAR 73	01:58:11	135		137	5	0	1	3613
INTAB	RELOCATABLE	13 MAR 73	01:59:15	136	2	57				3750
INTAB	FOR SYMB	13 MAR 73	01:59:16	137		66	5	0	1	3809
LIQRES	RELOCATABLE	13 MAR 73	01:59:19	138	1	4				3875
LIQRES	FOR SYMB	13 MAR 73	01:59:19	139		11	5	0	1	3880
LOCATE	RELOCATABLE	13 MAR 73	01:59:23	140	2	9				3891
LOCATE	FOR SYMB	13 MAR 73	01:59:23	141		13	5	0	1	3902
LPROPB	RELOCATABLE	13 MAR 73	01:59:26	142	1	6				3915
LPROPB	FOR SYMB	13 MAR 73	01:59:26	143		4	5	0	1	3922
LPROP	RELOCATABLE	13 MAR 73	01:59:29	144	2	13				3926
LPROP	FOR SYMB	13 MAR 73	01:59:30	145		8	5	0	1	3941
LWEGHT	RELOCATABLE	13 MAR 73	02:00:22	146	2	16				3949
LWEGHT	FOR SYMB	13 MAR 73	02:00:23	147		24	5	0	1	3967
MATHAX	RELOCATABLE	13 MAR 73	02:00:41	148	4	29				3991
MATHAX	FOR SYMB	13 MAR 73	02:00:42	149		30	5	0	1	4024
MIPE	RELOCATABLE	13 MAR 73	02:00:47	150	2	21				4054
MIPE	FOR SYMB	13 MAR 73	02:00:47	151		28	5	0	1	4077
NIENTH	RELOCATABLE	13 MAR 73	02:00:49	152	1	2				4105

NIENTH	FOR SYMB	13 MAR 73	02:00:49	153		2	5	0	1	4108
ONPROP	RELOCATABLE	13 MAR 73	02:00:52	154	2	7				4110
ONPROP	FOR SYMB	13 MAR 73	02:00:52	155		8	5	0	1	4119
OUTPUT	RELOCATABLE	13 MAR 73	02:01:29	156	2	21				4127
OUTPUT	FOR SYMB	13 MAR 73	02:01:29	157		13	5	0	1	4150
OXENTH	RELOCATABLE	13 MAR 73	02:01:34	158	1	2				4163
OXENTH	FOR SYMB	13 MAR 73	02:01:34	159		2	5	0	1	4166
PAGE	RELOCATABLE	13 MAR 73	02:01:43	160	2	7				4168
PAGE	FOR SYMB	13 MAR 73	02:01:43	161		31	5	0	1	4177
PARPMP	RELOCATABLE	13 MAR 73	02:01:52	162	3	27				4208
PARPMP	FOR SYMB	13 MAR 73	02:01:52	163		43	5	0	1	4238
PFNOB	RELOCATABLE	13 MAR 73	02:01:57	164	1	3				4281
PFNOB	FOR SYMB	13 MAR 73	02:01:57	165		2	5	0	1	4285
PFND	RELOCATABLE	13 MAR 73	02:02:01	166	2	12				4287
PFND	FOR SYMB	13 MAR 73	02:02:01	167		11	5	0	1	4301
PHIB	RELOCATABLE	13 MAR 73	02:02:04	168	1	2				4312
PHIB	FOR SYMB	13 MAR 73	02:02:04	169		3	5	0	1	4315
PROPB	RELOCATABLE	13 MAR 73	02:02:10	170	1	6				4318
PROPB	FOR SYMB	13 MAR 73	02:02:11	171		4	5	0	1	4325
PROP	RELOCATABLE	13 MAR 73	02:02:15	172	2	20				4329
PROP	FOR SYMB	13 MAR 73	02:02:15	173		21	5	0	1	4351
PSATH	RELOCATABLE	13 MAR 73	02:02:18	174	1	10				4372
PSATH	FOR SYMB	13 MAR 73	02:02:18	175		18	5	0	1	4383
PTDENS	RELOCATABLE	13 MAR 73	02:02:36	176	1	72				4401
PTDENS	FOR SYMB	13 MAR 73	02:02:37	177		70	5	0	1	4474
PTHEAT	RELOCATABLE	13 MAR 73	02:02:52	178	1	34				4544
PTHEAT	FOR SYMB	13 MAR 73	02:02:52	179		68	5	0	1	4579
PHTHON	RELOCATABLE	13 MAR 73	02:03:16	180	2	4				4647
PHTHON	FOR SYMB	13 MAR 73	02:03:16	181		6	5	0	1	4653
PVAPOR	RELOCATABLE	13 MAR 73	02:03:34	182	1	16				4659
PVAPOR	FOR SYMB	13 MAR 73	02:03:34	183		7	5	0	1	4676
RHOLIQ	RELOCATABLE	13 MAR 73	02:03:37	184	1	11				4683
RHOLIQ	FOR SYMB	13 MAR 73	02:03:37	185		6	5	0	1	4695
SPHSEG	RELOCATABLE	13 MAR 73	02:03:41	186	2	19				4701
SPHSEG	FOR SYMB	13 MAR 73	02:03:42	187		13	5	0	1	4722
SPHTDA	RELOCATABLE	13 MAR 73	02:03:53	188	1	75				4735
SPHTDA	FOR SYMB	13 MAR 73	02:03:54	189		83	5	0	1	4811
STOCON	RELOCATABLE	13 MAR 73	02:03:57	190	1	3				4894
STOCON	FOR SYMB	13 MAR 73	02:03:57	191		7	5	0	1	4898
TBOIL	RELOCATABLE	13 MAR 73	02:06:55	192	1	2				4905
TBOIL	FOR SYMB	13 MAR 73	02:06:55	193		2	5	0	1	4908
TCOND	RELOCATABLE	13 MAR 73	02:07:07	194	1	17				4910
TCOND	FOR SYMB	13 MAR 73	02:07:07	195		22	5	0	1	4928
TEL	RELOCATABLE	13 MAR 73	02:07:35	196	2	9				4950
TEL	FOR SYMB	13 MAR 73	02:07:36	197		15	5	0	1	4961
TEMP	RELOCATABLE	13 MAR 73	02:07:45	198	1	18				4976
TEMP	FOR SYMB	13 MAR 73	02:07:45	199		15	5	0	1	4995
THETAB	RELOCATABLE	13 MAR 73	02:07:48	200	1	2				5010
THETAB	FOR SYMB	13 MAR 73	02:07:48	201		3	5	0	1	5013
THKWTG	RELOCATABLE	13 MAR 73	02:08:09	202	2	16				5016
THKWTG	FOR SYMB	13 MAR 73	02:08:10	203		24	5	0	1	5034
TKGEOM	RELOCATABLE	13 MAR 73	02:08:28	204	3	20				5058
TKGEOM	FOR SYMB	13 MAR 73	02:08:28	205		20	5	0	1	5081
TIMELTB	RELOCATABLE	13 MAR 73	02:08:53	206	1	2				5101
TIMELTB	FOR SYMB	13 MAR 73	02:08:53	207		2	5	0	1	5104
TIMELT	RELOCATABLE	13 MAR 73	02:08:56	208	1	5				5106
TIMELT	FOR SYMB	13 MAR 73	02:08:57	209		4	5	0	1	5112
TNKWTA	RELOCATABLE	13 MAR 73	02:09:08	210	2	34				5112

TNKWTA	FOR SYMB	13 MAR 73	02:09:08	211		46	5	0	1	5152
TRAC	RELOCATABLE	13 MAR 73	02:09:15	212	1	2				5198
TRAC	FOR SYMB	13 MAR 73	02:09:15	213		2	5	0	1	5201
TSAT	RELOCATABLE	13 MAR 73	02:09:22	214	1	11				5203
TSAT	FOR SYMB	13 MAR 73	02:09:22	215		7	5	0	1	5215
TSATH	RELOCATABLE	13 MAR 73	02:09:32	216	1	10				5222
TSATH	FOR SYMB	13 MAR 73	02:09:33	217		18	5	0	1	5233
TSIZE1	RELOCATABLE	13 MAR 73	02:09:39	218	3	17				5251
TSIZE1	FOR SYMB	13 MAR 73	02:09:39	219		27	5	0	1	5271
TSTART	RELOCATABLE	13 MAR 73	02:09:41	220	1	2				5298
TSTART	FOR SYMB	13 MAR 73	02:09:41	221		3	5	0	1	5301
TURBN	RELOCATABLE	13 MAR 73	02:09:44	222	2	6				5304
TURBN	FOR SYMB	13 MAR 73	02:09:44	223		9	5	0	1	5312
TVPB	RELOCATABLE	13 MAR 73	02:09:46	224	1	2				5321
TVPB	FOR SYMB	13 MAR 73	02:09:46	225		2	5	0	1	5324
TVP	RELOCATABLE	13 MAR 73	02:09:48	226	2	8				5326
TVP	FOR SYMB	13 MAR 73	02:09:49	227		9	5	0	1	5336
VARNAM	RELOCATABLE	13 MAR 73	02:09:54	228	6	1				5345
VARNAM	FOR SYMB	13 MAR 73	02:09:54	229		6	5	0	1	5352
VENT	RELOCATABLE	13 MAR 73	02:10:01	230	2	44				5358
VENT	FOR SYMB	13 MAR 73	02:10:01	231		47	5	0	1	5404
VGVS	RELOCATABLE	13 MAR 73	02:10:03	232	2	6				5451
VGVS	FOR SYMB	13 MAR 73	02:10:04	233		12	5	0	1	5459
VPNB	RELOCATABLE	13 MAR 73	02:10:06	234	1	2				5471
VPNB	FOR SYMB	13 MAR 73	02:10:06	235		2	5	0	1	5474
VPN	RELOCATABLE	13 MAR 73	02:10:07	236	2	6				5476
VPN	FOR SYMB	13 MAR 73	02:10:07	237		5	5	0	1	5484
VPROPB	RELOCATABLE	13 MAR 73	02:10:10	238	1	6				5489
VPROPB	FOR SYMB	13 MAR 73	02:10:10	239		4	5	0	1	5496
VPROP	RELOCATABLE	13 MAR 73	02:10:12	240	2	9				5500
VPROP	FOR SYMB	13 MAR 73	02:10:12	241		6	5	0	1	5511
VSNDB	RELOCATABLE	13 MAR 73	02:10:14	242	1	3				5517
VSNDB	FOR SYMB	13 MAR 73	02:10:14	243		3	5	0	1	5521
VSD	RELOCATABLE	13 MAR 73	02:10:16	244	2	9				5524
VSD	FOR SYMB	13 MAR 73	02:10:16	245		8	5	0	1	5535
WFIND	RELOCATABLE	13 MAR 73	02:10:19	246	1	8				5543
WFIND	FOR SYMB	13 MAR 73	02:10:19	247		9	5	0	1	5552
WTACC	RELOCATABLE	13 MAR 73	02:10:22	248	2	4				5561
WTACC	FOR SYMB	13 MAR 73	02:10:22	249		10	5	0	1	5567
YLGINT	RELOCATABLE	13 MAR 73	02:10:25	250	1	16				5577
YLGINT	FOR SYMB	13 MAR 73	02:10:25	251		42	5	0	1	5594
YLNTRP	RELOCATABLE	13 MAR 73	02:10:27	252	1	1				5636
YLNTRP	FOR SYMB	13 MAR 73	02:10:27	253		3	5	0	1	5638
ZFIND	RELOCATABLE	13 MAR 73	02:10:31	254	2	35				5641
ZFIND	FOR SYMB	13 MAR 73	02:10:32	255		33	5	0	1	5678
ZGET	RELOCATABLE	13 MAR 73	02:10:40	256	1	2				5711
ZGET	FOR SYMB	13 MAR 73	02:10:40	257		2	5	0	1	5714
APUSUB	RELOCATABLE	14 MAR 73	22:09:49	258	4	28				5716
APUSUB	FOR SYMB	14 MAR 73	22:09:49	259		44	5	0	1	5748
CONTRL	RELOCATABLE	14 MAR 73	22:12:32	260	3	10				5792
CONTRL	FOR SYMB	14 MAR 73	22:12:32	261		13	5	0	1	5805
CNAMES	FOR PROC	28 MAR 73	00:05:08	262		3	1	0	1	5818
TANK	RELOCATABLE	28 MAR 73	00:14:24	263	4	161				5821
TANK	FOR SYMB	28 MAR 73	00:14:25	264		157	5	0	1	5986
CECLSS	FOR PROC	28 MAR 73	10:31:37	265		19	1	0	1	6143
CFUEL	FOR PROC	28 MAR 73	10:31:39	266		13	1	0	1	6162
CHPCAL	RELOCATABLE	28 MAR 73	10:32:04	267	6	90				6175
CHPCAL	FOR SYMB	28 MAR 73	10:32:04	268		126	5	0	1	6271

COMPIL	RELOCATABLE	28 MAR 73	10132152	269	6	168				6397
COMPIL	FOR SYMB	28 MAR 73	10132152	270		137	5	0	1	6571
STODTA	RELOCATABLE	28 MAR 73	10135143	271	4	75				6708
STODTA	FOR SYMB	28 MAR 73	10135144	272		103	5	0	1	6787
LSSCMP	RELOCATABLE	29 MAR 73	04124113	273	4	52				6890
LSSCMP	FOR SYMB	29 MAR 73	04124114	274		71	5	0	1	6946
OTRTNS	RELOCATABLE	29 MAR 73	04126102	275	7	141				7017
OTRTNS	FOR SYMB	29 MAR 73	04126103	276		127	5	0	1	7165
ECLSS	RELOCATABLE	06 APR 73	13132102	277	6	190				7292
ECLSS	FOR SYMB	06 APR 73	13132104	278		218	5	0	1	7488
FUELCL	RELOCATABLE	06 APR 73	13151152	279	5	217				7706
FUELCL	FOR SYMB	06 APR 73	13151156	280		260	5	0	1	7928
HEXF21	RELOCATABLE	06 APR 73	13152100	281	1	31				8188
HEXF21	FOR SYMB	06 APR 73	13152100	282		15	5	0	1	8220
NEXT AVAILABLE LOCATION-										8235

ASSEMBLER PROCEDURE TABLE EMPTY

COBOL PROCEDURE TABLE EMPTY

FORTRAN PROCEDURE TABLE

D NAME	LOCATION	LINK	D NAME	LOCATION	LINK	D NAME	LOCATION	LINK
CACCU	50178	1	CAPU	50570	2	CCNFIG	51970	3
CCNTRL	53174	4	CDCYCL	53454	5	CECLSS	172006	265
CENG	53734	6	CFLRAT	54014	7	CFLUID	54098	8
CFUEL	172538	266	CHEX	54126	9	CHSORC	54854	11
CHTX	54742	10	CIOUNT	55218	12	CKEYS	55358	13
CMATRL	55386	14	CMOTOR	55862	15	CNAMES	162906	262
CONST	55918	16	CPAGE	55974	17	CPUMP	56198	18
CSYSHT	56758	19	CTAB	56926	20	CTABA	57206	21
CTANK	57262	22	CTURBN	57794	23	DUMMY	58158	24
SPUMP	58242	25	TABLOK	58298	26	TANKWT	58326	27

ENTRY POINT TABLE EMPTY

Appendix B
THE CRYOGENIC INTEGRATED MATH MODEL
(TCIMM)

PART III - CROSS REFERENCE OF PROGRAM FILE

It is often of interest to know which subprograms call a specific routine in a given program sequence. This kind of information for program file TCIMM is presented in the following pages. An explanation of the XREF processes which generates the cross reference listing is given below.

XREF (Cross Reference Listing of Relocatable Elements): The XREF processor generates a cross reference listing of all entry points and undefined symbols in the specified program file that has been PACKed and PREP'D. The names of the relocatable elements are listed alphabetically. Beside each element name, the names of the element entry points are listed. Beside each entry point name, the names of all relocatable elements in the program file which reference this entry point are listed. An element entry point is the result of an assembly or compilation and specifies the location at which execution of the program element commences. A compiled FORTRAN V subroutine or function has one entry point corresponding to the name of the subroutine or function. An undefined symbol (or external reference) is the result of a subroutine call or reference to an array not contained within the element. Any external FORTRAN reference creates an undefined symbol.

CROSS REFERENCE OF FILE TCIMM.

BXREF*XRREF,XREF

ACCRES	01	(000037)	(ACCRES) ,CRYCON
ACQWT	01	(000034)	(ACQWT) ,CRYCON
AFUNC	01	(000446)	(TKGEOM) ,TNKHTA
ALPHAB	01	(000007)	(ALPHAB)
APUFLO	01	(000337)	(APUFLO) ,CONSUM
APUSUB	01	(000707)	(APUSUB) ,CRYCON
APUSUP	01	(003175)	(APUSUP) ,CRYCON
ARACYL	01	(000617)	(GOMTRY) ,TKGEOM,TNKHTA
AREAFR	01	(000660)	(GOMTRY) ,TKGEOM
ARSPHR	01	(000727)	(GOMTRY) ,TKGEOM,TNKHTA
BETAB	01	(000056)	(BETAB) ,ECLSS,FUELCL
CFTW	01	(000131)	(CFTW) ,CMPCAL,LSSCMP
CMPCAL	01	(002673)	(CMPCAL) ,CRYCON
COMFLO	01	(000263)	(COMFLO) ,CMPCAL,LSSCMP
COMPIL	01	(005142)	(COMPIL) ,CONTRL
CONE	01	(000245)	(GOMTRY)
CONSUM	01	(000027)	(CONSUM) ,CRYCON
CPHI	01	(000103)	(CPHI) ,VPROP
CPIG	01	(000103)	(CPIG) ,CPVTD
CPSI	01	(000113)	(CPSI) ,VPROP
CPVTD	01	(000157)	(CPVTD) ,VSND,CPVTDB
CPVTDB	01	(000040)	(CPVTDB) ,PHTON,CSUBPV
CRYCON	01	(000202)	(CRYCON) ,CONTRL
CSPF21	01	(000063)	(CSPF21) ,FUELCL
CSUBP	01	(000151)	(CSUBP) ,HEXELC,VGVS,APUSUB,COMFLO,APUSUP,ECLSS,FUELCL
CSUBPV	01	(000056)	(CSUBPV) ,CSUBV,CSUBP
CSUBPI	01	(000115)	(CSUBPI) ,GASGEN,FLORAT,APUSUB,CMPCAL,APUSUP,APUFLO
CSUBV	01	(000113)	(CSUBV) ,VGVS,TANK,COMFLO,ECLSS,APUSUP,FUELCL
CYLHED	01	(000364)	(SPHSEG) ,TKGEOM,TNKHTA
CYLHNR	01	(000306)	(GOMTRY) ,TKGEOM
CYLSPH	01	(000347)	(GOMTRY) ,TKGEOM
CYHSPH	01	(000451)	(SPHSEG) ,TKGEOM
DATAN2	01	(000257)	(DATAN2) ,PHTON,ONPROP,DENSON,CSUBPV,BETAB
DATA02	01	(000265)	(DATA02) ,PHTON,ONPROP,DENSON,CSUBPV,BETAB
DCALC	01	(000274)	(DCALC) ,DFND
DENSON	01	(000052)	(DENSON) ,LSSCMP,ECLSS,APUSUP,FUELCL
DFND	01	(000211)	(DFND) ,PROP,LPROP,VPROP,VSND,DFNDB
DFNDB	01	(000034)	(DFNDB) ,DENSON,CSUBPV
DIAG	01	(000062)	(DIAG) ,HIPE,LWEIGHT,LOCATE,TSIZEI,VENT,GETCON,FLORAT,FINTAB,ENGINE,TANK,CRYCON,COMFLO
DPDD	01	(000327)	(DPDD) ,VSND,DPDDB,CPVTD
DPDDB	01	(000034)	(DPDDB) ,PHTON,BETAB
DPDT	01	(000271)	(DPDT) ,DPDTB,CPVTD
DPDTB	01	(000027)	(DPDTB) ,PHTON,BETAB
DPDTVP	01	(000101)	(DPDTVP) ,LPROP,TVP
DSATL	01	(000060)	(DSATL) ,DFND
DSATV	01	(000064)	(DSATV) ,DFND
ECLSS	01	(004406)	(ECLSS) ,CRYCON
ELIPSG	01	(000324)	(SPHSEG) ,TKGEOM
ENGINE	01	(000310)	(ENGINE) ,CONSUM
ENTHOH	01	(000560)	(MATHAX) ,HEATEX,FLORAT
FDNSTY	01	(000576)	(MATHAX) ,THKHTG,TSIZEI
FINDR	01	(000014)	(FINDR) ,MATHAX,VENT,VGVS,TANK,COMFLO,APUSUP
FING1	01	(000322)	(FING1) ,LPROP,VPROP
FING2	01	(000314)	(FING2) ,LPROP,VPROP
FING3	01	(000321)	(FING3) ,CPVTD

FINTAB	01 (000163)	(FINTAB) ,PARPHP,MATHAX,LWEGHT,THKWTG,HEXELC,VENT,HEATEX,ENGINE,TANK,CMPCAL,ECLSS,APUSUP ,FUELCL,APUFLO
FLODER	01 (000041)	(FLODER) ,COMFLO
FLORAT	01 (000531)	(FLORAT) ,CONSUM
FRONE	01 (000422)	(GOMTRY) ,TKGEOM
FRIEAD	01 (000422)	(SPHSEG) ,TKGEOM
FUELCL	01 (005355)	(FUELCL) ,CRYCON
GASGEN	01 (000211)	(GASGEN) ,CMPCAL,APUSUP
GETCON	01 (000053)	(GETCON) ,CMPCAL,LSSCMP,OTRTNS
GSDNST	01 (000623)	(MATHAX) ,VENT,APUSUB,TANK,CMPCAL,FUELCL,ACCRES
GSZDMS	01 (000650)	(MATHAX) ,VENT,TANK
HEATEX	01 (001742)	(HEATEX) ,TANK,CMPCAL,APUSUP
HEXELC	01 (000263)	(HEXELC) ,ECLSS
HEXF21	01 (000751)	(HEXF21) ,FUELCL
HFUNC	01 (000504)	(TKGEOM) ,TNKWTG
HPTCP	01 (000014)	(HPTCP) ,CSUBP
HPTCV	01 (000014)	(HPTCV) ,CSUBV
HPTGAM	01 (000014)	(HPTGAM) ,COMFLO
HPW	01 (000006)	(HPW)
HSPHER	01 (000471)	(GOMTRY) ,TKGEOM
HTLEAK	01 (000006)	(HTLEAK)
HVAP	01 (000110)	(HVAP)
HYENTH	01 (000621)	(HYENTH) ,MATHAX,APUSUB,APUSUP,FUELCL
INTAB	01 (001510)	(INTAB) ,CONTRL
LIQRES	01 (000063)	(LIQRES) ,CRYCON
LOCAT	01 (000167)	(LOCATE) ,MIPE
LPROP	01 (000254)	(LPROP) ,PROP,LPROPB,CPVTD
LPROPB	01 (000073)	(LPROPB)
LSSCMP	01 (001452)	(LSSCMP) ,ECLSS
LWEGHT	01 (000353)	(LWEGHT) ,CMPCAL,LSSCMP
MIPE	01 (000506)	(MIPE) ,PARPHP,THKWTG,MATHAX,LWEGHT,HEXELC,VENT,HEATEX,ENGINE,TANK,CMPCAL,ECLSS,FUELCL ,APUSUP,APUFLO
NIENTH	01 (000022)	(NIENTH) ,ECLSS
OAPUSB	01 (003773)	(OTRTNS) ,APUSUB
OAPUSP	01 (003766)	(OTRTNS) ,APUSUP
ONPROP	01 (000146)	(ONPROP) ,OXENTH,NIENTH,ZGET
OPAPUF	01 (003761)	(OTRTNS) ,APUFLO
OPTPOW	01 (003722)	(OTRTNS) ,ECLSS
OTPACC	01 (003742)	(OTRTNS) ,WTACC
OTPAO	01 (003745)	(OTRTNS) ,ACGWT
OTPFLT	01 (003753)	(OTRTNS) ,FLORAT
OTPFIX	01 (003756)	(OTRTNS) ,FLORAT
OTPHX	01 (003711)	(OTRTNS) ,CMPCAL,APUSUP
OTPHXE	01 (003717)	(OTRTNS) ,ECLSS
OTPHXF	01 (003714)	(OTRTNS) ,FUELCL
OTPPMP	01 (003725)	(OTRTNS) ,CMPCAL
OTPTRB	01 (003730)	(OTRTNS) ,CMPCAL
OTPTSZ	01 (003733)	(OTRTNS) ,TSIZEI
OTPHSM	01 (003750)	(OTRTNS) ,CRYCON
OTUNIT	01 (000512)	(OUTPUT) ,CONTRL
OUTPA	01 (000425)	(OUTPUT) ,OTRTNS
OUTPF	01 (000344)	(OUTPUT) ,OTRTNS
OUTPFI	01 (000371)	(OUTPUT) ,OTRTNS
OUTPI	01 (000402)	(OUTPUT) ,OTRTNS
OUTPW	01 (000450)	(OUTPUT) ,OTRTNS
OXENTH	01 (000022)	(OXENTH) ,MATHAX,APUSUB,ECLSS,APUSUP,FUELCL
PAGE	01 (000121)	(PAGE) ,INTAB,ENGINE,CONTRL,DIAG,TANK,CMPCAL,COMPIL,LSSCMP,OTRTNS,ECLSS,FUELCL
PARPHP	01 (000600)	(PARPHP) ,CMPCAL

PFND	01 (000251) (PFND) ,PFNDB,LPROP,VPROP,DCALC,CPVTD
PFNDB	01 (000030) (PFNDB)
PHIB	01 (000012) (PHIB) ,PHTHON
PHTHON	01 (000100) (PHTHON) ,ECLSS,APUSUP,FUELCL
PROP	01 (000374) (PROP) ,PROPB
PROPB	01 (000073) (PROPB) ,ONPROP
PSATH	01 (000122) (PSATH) ,APUSUB
PTDENS	01 (000551) (PTDENS) ,ZFIND
PTHEAT	01 (000671) (PTHEAT) ,HPTCP,HPTCV,HPTGAM
PUMPEF	01 (000522) (MATHAX) ,PARPMP
PVAPOR	01 (000316) (PVAPOR) ,ZFIND
RHOLIQ	01 (000203) (RHOLIQ) ,MATHAX,VENT,FLORAT,APUSUB,TANK,CMPCAL
RVHIPE	01 (000521) (HIPE)
SPACE	01 (000507) (OUTPUT) ,OTRTNS
SPHERE	01 (000544) (GOMTRY) ,TNKWTA
SPHSEG	01 (000266) (SPHSEG) ,TNKWTA
STOCON	01 (000040) (STOCON) ,COMPII
STODTA	01 (000006) (STODTA) ,CONTRL
TANK	01 (003716) (TANK) ,CRYCON
TBOIL	01 (000014) (TBOIL) ,HVAP
TCOND	01 (000371) (TCOND) ,APUSUB,APUSUP,ECLSS,FUELCL
TCRCAL	01 (000527) (MATHAX) ,HEATEX
TCRCLC	01 (000700) (MATHAX) ,HEATEX
TCRLW	01 (000735) (MATHAX) ,HEATEX
TCRRAZ	01 (000746) (MATHAX) ,HEATEX
TEL	01 (000161) (TEL) ,HIPE
TEMP	01 (000321) (TEMP) ,TMELT
THETAB	01 (000012) (THETAB) ,PHTHON
THKWTG	01 (000323) (THKWTG) ,TNKWTA
TMELT	01 (000077) (TMELT) ,TMELTB
TMELTB	01 (000021) (TMELTB)
TNKWTA	01 (001057) (TNKWTA) ,TSIZEI,WTACC
TRAC	01 (000013) (TRAC)
TSAT	01 (000234) (TSAT) ,HEATEX,VENT,TANK
TSATH	01 (000122) (TSATH)
TSIZEI	01 (000416) (TSIZEI) ,CRYCON
TSTART	01 (000014) (TSTART) ,HVAP
TURBN	01 (000105) (TURBN) ,CMPCAL
TVP	01 (000136) (TVP) ,TVPB
TVPB	01 (000020) (TVPB)
VARNAM	01 (000004) (VARNAM)
VENT	01 (001105) (VENT) ,TANK
VFUNC	01 (000410) (TKGEOM) ,TNKWTA
VGVS	01 (000106) (VGVS) ,CMPCAL,LSSCMP
VPN	01 (000104) (VPN) ,PROP,LPROP,TVP,VPNB,VSND,DFND
VPNB	01 (000020) (VPNB)
VPROP	01 (000157) (VPROP) ,PROP,LPROP,VPROPB
VPROPB	01 (000073) (VPROPB)
VSND	01 (000145) (VSND) ,VSNDB
VSNDB	01 (000036) (VSNDB)
WFIND	01 (000124) (WFIND) ,TEMP
WUACL	01 (000726) (MATHAX) ,HEATEX
WTACC	01 (000062) (WTACC) ,CRYCON
YLGINT	01 (000405) (YLGINT) ,TEL
YLNTRP	01 (000007) (YLNTRP)
ZFIND	01 (000436) (ZFIND) ,MATHAX,VENT,APUSUB,TANK,APUSUP,FUELCL
ZGET	01 (000022) (ZGET) ,FUELCL,APUSUP,ECLSS,APUSUB
DONE	